

Initial framework for engineering-scale statistical creep-fatigue modeling

Applied Materials Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free at OSTI.GOV (<http://www.osti.gov/>), a service of the U.S. Dept. of Energy's Office of Scientific and Technical Information

Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312
www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: **orders@ntis.gov**

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI)

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: **reports@osti.gov**

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

Initial framework for engineering-scale statistical creep-fatigue modeling

Applied Materials Division
Argonne National Laboratory

September 2021

Prepared by

A. Rovinelli, Argonne National Laboratory
A. Venkataraman, Argonne National Laboratory
M. C. Messner, Argonne National Laboratory

Abstract

This report describes the integration of new solid and interface-cohesive mechanics systems into MOOSE. The purpose of these new systems is to support the ability of MOOSE to run full-field crystal plasticity finite element method simulations of key material processes in high temperature metallic materials. These simulations could be used to help accurately predict the performance of key high temperature structural materials in future advanced nuclear reactor components. Previous work implemented preliminary versions of many of these systems in MOOSE Apps. The current work reports on their integration into the main MOOSE tensor mechanics module along with associated improvements to the basic formulations and numerical implementations. Finally, the report provides an example of the full-field crystal plasticity simulations now possible in MOOSE, including examples of realistic geometries requiring millions of degrees of freedom to resolve the microstructural features and macroscale geometry.

Table of Contents

Abstract	i
Table of Contents	iii
List of Figures	v
List of Tables	vii
1 Introduction	1
1.1 Purpose	1
1.2 Organization	1
2 New Solid Mechanics Kernels and Material System	3
2.1 Solid mechanical kernel formulation	3
2.1.1 Total Lagrangian theory	4
2.1.2 Updated Lagrangian theory	5
2.2 Material system	7
2.2.1 Kinematics	7
2.2.2 Stress update options	8
2.3 \bar{F} stabilization	12
2.3.1 The need for stabilization	12
2.3.2 \bar{F} and \bar{B} Stabilization	12
2.3.3 Implementation of the \bar{F} method	13
2.3.4 Cook's membrane	14
2.4 Homogenization subsystem	14
2.4.1 Mathematical description	16
2.4.2 Example simulation	16
3 Integrating the Interface-Cohesive System into MOOSE	19
3.1 Large deformation Cohesive zone modeling approach	21
3.1.1 The interface deformation gradient	21
3.1.2 Interface Coordinate Systems	22
3.1.3 Total Lagrangian Residual	23
3.2 The cohesive zone material system	26
3.2.1 Cohesive zone constitutive models	26
3.2.2 The displacement jump provider	26
3.3 Integral PK1 traction formulation	27
3.3.1 Derivative of T with respect to the discrete displacements	27
3.3.2 The small strain formulation	28
4 Full-scale CPFEM Simulations of Cyclic Load	31
4.1 Microstructure and loading conditions	31
4.2 Results and Analysis	34
5 Conclusions and Future Work	41
Acknowledgments	43
Bibliography	45

List of Figures

1.1	Mesh for a full-field simulation of a cylindrical test section. The model represents deformation and failure in the grain bulk with crystal plasticity and deformation and failure on the grain boundaries with an interface-cohesive model. The entire simulation runs in MOOSE using large deformation kinematics to track geometrical effects caused by grain rotations and the eventual large deformations and necking in the sample. This example model has about 1.5 million quadratic tetrahedral elements.	2
2.1	Reference frames used in deriving the kernel formulations.	4
2.2	Plot of σ_{yy} and σ_{zz} in the cube as it rotates, for different numbers of time steps.	11
2.3	Shear stress/shear strain plot comparing the Truesdell and Jaumann rates for very large shear deformations.	12
2.4	Cook's membrane: a reference problem for testing locking and stabilization strategies.	15
2.5	Demonstration of \bar{F} stabilization on a small deformation problem.	15
2.6	Demonstration of \bar{F} stabilization on a large deformation problem.	16
2.7	Simple, 2D large deformation homogenization cell problem.	17
2.8	Results showing that the controlled deformation values match their targets.	17
2.9	Results showing that the controlled stress values match their targets.	18
3.1	Schematic depicting the objects required for cohesive zone modeling in <i>MOOSE</i> , and the objects automatically added by the <i>CohesizeZoneMaster</i> action	20
3.2	Schematic showing the bulk material surfaces (blue lines) and the interface midplane (dashed black lines) in the deformed and undeformed configuration and the associated deformation gradients. The figure also shows how the interface coordinate system evolves according to the deformation of interface midplane (green arrows) Note: in the undeformed configuration all surfaces coincide, however they are shown separated for clarity.	21
3.3	a) Interface coordinate system, b) interface coordinate system in global coordinates for the undeformed configuration, and c) the interface coordinate system in the deformed configuration. Q_0 is rotation matrix transforming from the interface coordinate system to the global coordinate system in the undeformed configuration. The rotation matrix R rotates a vector from the deformed to the undeformed configuration. The variables a and A are the interface areas in the deformed and undeformed configuration, respectively.	23
3.4	Stretch (simulation time 0-1) plus rotation (90° around the y-axis, simulation time 1-2 test using a linear traction separation law implemented using the total and incremental formulation.	29
4.1	(a) A representative meshed microstructure used in the present work. This microstructure has 1000 grains and ~ 1.5 M elements. (b) Schematic of the creep loading conditions applied on the microstructure. Maximum stress of 100 MPa was used across all the simulations.	32

4.2	(a) Finite element mesh resulting from the old method using Gmsh and (b) mesh resulting from the updated method (using Coreform Trelis).	33
4.3	The geometries simulated in the present study: full cylinder, quarter cylinder, cube, and cube geometries with larger aspect ratios.	35
4.4	A compilation of the creep curves for the different geometries simulated are shown here. The cylinder microstructure reaches tertiary creep earlier than the cubic microstructure.	35
4.5	(a) The inelastic strain distribution at the end of the creep simulations in the full cylinder microstructure. (b) The corresponding distribution of the damage metric D in the cylinder microstructure.	36
4.6	Comparison of average Poisson's ratio for the different simulated geometries.	36
4.7	Comparison of the equivalent strains partitioned between the grain and grain boundaries (GB) for the different microstructures simulated.	37
4.8	Comparison of the grain boundary sliding strains partitioned between the axial and transverse directions for the different microstructures simulated. .	38
4.9	Comparison of the grain boundary opening strains partitioned between the axial and transverse directions for the different microstructures simulated. .	38
4.10	(a) Homogenized strains (homogenized over the entire domain of the microstructure) partitioned between the grain and grain boundaries (GB) for the different microstructures simulated.	39

List of Tables

2.1	Basic kinematic measure provided by the material system.	8
2.2	Options for implementing new material models	8
4.1	The material parameters used in the present simulations.	34

1 Introduction

1.1 Purpose

This report describes the integration of new solid mechanics and interface mechanics systems into the MOOSE Tensor Mechanics module, aimed at providing a capability to simulate large-scale, finite deformation, crystal plasticity finite element simulations. Figure 1.1 shows an example simulation of this type.

Previous research at ANL developed the basic capabilities to perform simulations of this kind in a separate MOOSE app called DEER [1]. These simulations consist of three main modules

1. A solid mechanics formulations with the correct large-deformation Jacobian for represent grain bulk deformation.
2. A large deformation interface-cohesive formulation for representing creep cavitation on grain boundaries.
3. A homogenization constraint system for imposing cell average stress or strain constraints to determine effective macroscale mechanical properties.

Past work at ANL developed the theory behind these modules [1–6]. The main objective of this develop work was to improve the underlying implementations and integrate the system into the mainline MOOSE. However, in the process of integrating the modules the development of several significant, new systems was required:

- A framework for maintaining both updated and total Lagrangian mechanics kernels.
- A new material system, aimed at making the task of implementing a new constitutive model into MOOSE much easier.
- Stabilization for linear hex and quad elements for nearly-incompressible problems.
- Improvements to the interface-cohesive modeling framework both to improve its numerical performance and to better integrate it into the tensor mechanics module.

1.2 Organization

Chapter 2 describes the new solid mechanics system integrated into MOOSE, including a description of the theory behind the basic formulation, the material system, the stabilization system, and the homogenization subsystem. Chapter 3 provides similar background for the large deformation interface-cohesive modeling system. Chapter 4 then demonstrates the new capabilities by summarizing work on full-scale, massively-parallel Crystal Plasticity Finite Element Method (CPFEM) simulations of cyclic creep-fatigue loading of metallic materials. The final chapter then discusses future work on the new mechanics systems and on full-scale CPFEM simulations in MOOSE.

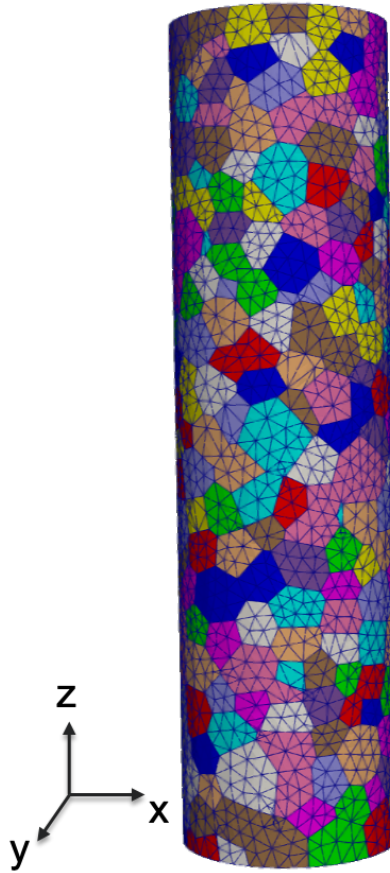


Figure 1.1: Mesh for a full-field simulation of a cylindrical test section. The model represents deformation and failure in the grain bulk with crystal plasticity and deformation and failure on the grain boundaries with an interface-cohesive model. The entire simulation runs in MOOSE using large deformation kinematics to track geometrical effects caused by grain rotations and the eventual large deformations and necking in the sample. This example model has about 1.5 million quadratic tetrahedral elements.

2 New Solid Mechanics Kernels and Material System

This chapter addresses the first objective detailed in Chapter 1: transfer ANL work on the solid mechanics formulation in MOOSE to the mainline MOOSE framework (in the Tensor Mechanics module). This work ended up requiring a significant amount of additional development effort to support more general use of the kernels, material system, homogenization system, and ancillary objects. The new kernel system includes:

1. Both total and updated Lagrangian formulations of the stress equilibrium residual for large deformation problems.
2. A comprehensive new material system for the kernels, which allows the user to implement models using any convenient stress and strain measures, automatically converting from the user-provided stress and algorithmic tangent measures to the measures required by the kernel formulations.
3. \bar{F} stabilization for nearly incompressible problems using linear hex or quad elements.
4. A complete transfer of a system for imposing cell-volume-average homogenization constraints from the ANL DEER application into mainline MOOSE. This included modifications to ensure the system works with the new stabilization and material systems.

While MOOSE already has a large deformation solid mechanics kernel, these kernels do not have the correct Jacobian for large deformations or for stabilized problems using linear elements. This significantly reduces the nonlinear convergence rate of large deformation problems, making these simulations substantially more computationally expensive than they need to be. The overall goal of the present work is to provide a large deformation solid mechanics system with the correct, exact Jacobian.

A previous ANL technical report [1] outlines much of the theory underlying these modifications. The following chapter summarizes the important aspects of new solid mechanics system, focusing on the new elements not described our previous report. The descriptions here also serve as the basis of the MOOSE documentation supporting the new mechanics system.

2.1 Solid mechanical kernel formulation

The new kernel system provides two options for enforcing stress equilibrium over a domain described by a map between the reference configuration of the body, defined as Ω_0 with coordinates \mathbf{X} , and the current configuration of the body defined as Ω with coordinates \mathbf{x} . The mathematics here use upper case letters to define quantities related to the reference configuration and lower case letters to define quantities related to the current configuration. Specifically,

$$f_{i,j} = \frac{\partial f_i}{\partial x_j} \quad (2.1)$$

denotes a spatial gradient with respect to the current coordinates while

$$f_{i,J} = \frac{\partial f_i}{\partial X_j} \quad (2.2)$$

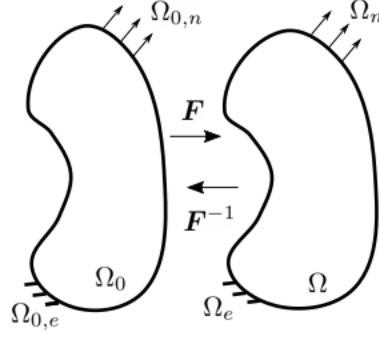


Figure 2.1: Reference frames used in deriving the kernel formulations.

denotes a gradient with respect to the reference coordinates. Figure 2.1 describes these two frames of reference.

2.1.1 Total Lagrangian theory

The total Lagrangian theory enforces the equilibrium condition mapped back to the reference configuration, weakly solving the differential equation:

$$P_{iJ,J} + b_i = 0 \text{ on } \Omega_0 \quad (2.3)$$

$$P_{iJ}N_j = \hat{t}_i \text{ on } \partial\Omega_{0,n} \quad (2.4)$$

$$u_i = \hat{u}_i \text{ on } \partial\Omega_{0,e} \quad (2.5)$$

where P_{iJ} is the 1st Piola-Kirchhoff stress, b_i are the body forces in the updated configuration, N_j are the boundary normals in the reference configuration, \hat{t}_i are the tractions in the updated configuration, u_i is the displacement field, and \hat{u}_i are imposed displacement boundary conditions.

This strong differential equation transforms to the weak residual

$$R^\alpha = \int_V P_{iK} \phi_{i,K}^\alpha dV \quad (2.6)$$

with the corresponding Jacobian

$$J^{\alpha\beta} = \int_V \phi_{i,J}^\alpha T'_{iJkL} G_{kL}^\beta dV \quad (2.7)$$

where P_{iK} is the first Piola-Kirchhoff stress,

$$T'_{iJkL} = \frac{dP_{iJ}}{dF_{kL}} \quad (2.8)$$

$\phi_{i,J}^\alpha$ are the test function gradients (with respect to the reference coordinates) and

$$G_{iJ}^\beta = \frac{dF_{iJ}}{dY^\beta} \quad (2.9)$$

with Υ^β the discrete (nodal) displacements. For the unstabilized case

$$G_{ij}^\beta = \psi_{i,J}^\beta \quad (2.10)$$

with $\psi_{i,J}^\beta$ the trial function gradients with respect to the reference coordinates.

2.1.2 Updated Lagrangian theory

The updated Lagrangian theory enforces the equilibrium condition on the current configuration:

$$\sigma_{ij,j} + b_i = 0 \text{ on } \Omega \quad (2.11)$$

$$\sigma_{ij} n_j = \hat{t}_i \text{ on } \partial\Omega_n \quad (2.12)$$

$$u_i = \hat{u}_i \text{ on } \partial\Omega_e \quad (2.13)$$

where σ_{ij} is the Cauchy stress.

For large deformation kinematics the weak form is

$$R^\alpha = \int_v \sigma_{ik} \phi_{i,k}^\alpha dv \quad (2.14)$$

with the corresponding Jacobian

$$J^{\alpha\beta} = \int_v \left\{ T_{ijkl} \phi_{i,j}^\alpha f_{km} g_{ml}^\beta + \sigma_{ij} \left(\phi_{k,k}^\alpha \psi_{ij}^\beta - \phi_{k,j}^\alpha \psi_{ik}^\beta \right) \right\} dv \quad (2.15)$$

where σ_{ik} is the Cauchy stress,

$$T_{ijkl} = \frac{\partial \sigma_{ij}}{\partial \Delta l_{kl}} \quad (2.16)$$

with

$$\Delta l_{kl} = \Delta F_{kM} F_{Ml}^{-1} \quad (2.17)$$

the incremental spatial velocity gradient, $\phi_{i,j}^\alpha$ are the test function gradients (with respect to the current coordinates) and

$$g_{ij}^\beta = \frac{dF_{iK}}{d\Upsilon^\beta} F_{Kj}^{-1} \quad (2.18)$$

with Υ^β the discrete (nodal) displacements. For the unstabilized case

$$g_{ij}^\beta = \psi_{i,j}^\beta \quad (2.19)$$

with $\psi_{i,j}^\beta$ the trial function gradients with respect to the current coordinates.

2.1.2.1 Small deformation theory

For small deformations both the updated and total Lagrangian formulation degenerate to the same mathematical system:

$$s_{ij,j} + b_i = 0 \text{ on } \Omega \quad (2.20)$$

$$s_{ij}n_j = \hat{t}_i \text{ on } \partial\Omega_n \quad (2.21)$$

$$u_i = \hat{u}_i \text{ on } \partial\Omega_e \quad (2.22)$$

where s_{ij} is the small (engineering) stress and there is now no difference between the current and reference configurations.

The residual and Jacobian degenerate to

$$R^\alpha = \int_V s_{ij} \phi_{i,j}^\alpha dv \quad (2.23)$$

and

$$J^{\alpha\beta} = \int_V \phi_{i,j}^\alpha C_{ijkl} g_{kl}^\beta dV \quad (2.24)$$

for the small deformation case, with s_{ij} the small stress,

$$C_{ijkl} = \frac{\partial s_{ij}}{\partial \varepsilon_{kl}} \quad (2.25)$$

with ε_{kl} the small strain and

$$g_{kl}^\beta = \psi_{k,l}^\beta \quad (2.26)$$

for the unstabilized case.

2.1.2.2 Discussion on the kernel formulations

It is more natural and common to define the constitutive response for the total Lagrangian formulation as the 1st Piola Kirchhoff stress as a function of the deformation gradient, $P_{iJ}(F_{kL})$ updated Lagrangian formulation as the Cauchy stress as a function of the deformation gradient, $\sigma_{ij}(F_{kL})$, with the deformation gradient defined as

$$F_{iJ} = \delta_{i,J} + u_{i,J} \quad (2.27)$$

with δ the Kronecker Delta. However, the Cauchy stress and the 1st Piola Kirchhoff stress are related

$$P_{iK} = J \sigma_{ij} F_{Kj}^{-1} \quad (2.28)$$

and so it is possible to convert a native Cauchy stress constitutive model to the 1st Piola-Kirchhoff stress and vice versa.

If the boundary conditions, body force, and constitutive model are all identical then the updated and total Lagrangian formulations will return exactly the same results. There is no difference in the final results when using a updated Lagrangian or total Lagrangian model.

However, at times one formulation may be more convenient than another. For example, the homogenization system only works with the total Lagrangian kernel because of

the difficulty in including the extra homogenization field in the kinematic spatial derivatives. The current MOOSE Tensor Mechanics system uses an updated Lagrangian formulation and so new system also includes both theories for backward compatibility.

In theory it can be more efficient to couple a “native” Cauchy stress constitutive model to the updated Lagrangian configuration and a “native” 1st Piola-Kirchhoff model to the total Lagrangian configuration. However, the currently-implemented material system always converts Cauchy stress to 1st Piola-Kirchhoff stress and vice-versa so that models can be used with either the updated or total Lagrangian kernels.

2.2 Material system

The material system for the new solid mechanics system consists of:

1. A strain calculator to provide basic kinematic quantities for mapping configurations in the kernel and defining the stress update.
2. A stress update object, which in the end must provide both the Cauchy and 1st Piola Kirchhoff stress measures along with the associated algorithmic tangents
3. Optionally, additional material objects part of the homogenization system.

The objective of these objects is to provide the stress update needed by *both* the updated and total Lagrangian formulation kernels. However, the user only needs to define the “main” stress measure and an associated derivative, listed in Table 2.2 below. The base class code then takes care of translating that stress and associated derivative into the “missing” stress and derivative.

2.2.1 Kinematics

It does not matter which kinematic quantities the user elects to use in the defining the stress update model, so long as the model returns the correct stress measure and associated derivative. Table 2.1 lists the basic kinematic tensors provided by the material system. Some of the stress update model base class options provide additional kinematic measures, described in Table 2.2.

The strain calculator also subtracts any eigenstrain increments from the total strain increment to define the mechanical strain. Furthermore, it adds in the extra gradient associated with the homogenization system, described below.

The strain calculator uses the spatial velocity gradient, defined as:

$$\Delta l_{ij} = \delta_{ij} - f_{ij}^{-1} \quad (2.29)$$

for large deformations and

$$\Delta l_{ij} = F_{ij}^{(new)} - F_{ij}^{(old)} \quad (2.30)$$

for small deformations, to define the incremental strains. Table 2.1 provides the definition of the other standard kinematic variables.

Table 2.1: Basic kinematic measure provided by the material system.

Quantity name	Definition, large kinematics	Definition, small kinematics
Deformation gradient	$F_{iJ} = \delta_{iJ} + \frac{\partial u_i}{\partial X_J}$	$F_{ij} = \delta_{ij} + \frac{\partial u_i}{\partial x_j}$
Inverse deformation gradient	F_{Ji}^{-1}	δ_{ji}
Inverse incremental deformation gradient	$f_{ij}^{-1} = F_{iK}^{(old)} F_{Kj}^{(new)-1}$	$f_{ij} = \delta_{ij}$
Volume change	$J = \det F$	$J = 1$
Total strain increment	$\Delta d_{ij} = \frac{1}{2} (l_{ij} + l_{ji})$	Same
Mechanical strain increment	$\Delta \varepsilon_{ij} = \Delta d_{ij} - \Delta \varepsilon_{ij}^{(eigen)}$	Same
Total strain	$d_{ij}^{(new)} = d_{ij}^{(old)} + \Delta d_{ij}$	Same
Mechanical strain	$\varepsilon_{ij} = \varepsilon_{ij}^{(old)} + \Delta \varepsilon_{ij}$	Same

Table 2.2: Options for implementing new material models

Description	Extra kinematic tensors	Stress measure	Required Jacobian
Cauchy stress material	None	Cauchy stress σ_{ij}	$\frac{d\sigma_{ij}}{d\Delta l_{kl}}$
1st Piola-Kirchhoff stress material	None	1st Piola-Kirchhoff stress P_{iJ}	$\frac{dP_{iJ}}{dF_{kL}}$
2nd Piola-Kirchhoff stress material	Green-Lagrange strain $E_{IJ} = \frac{1}{2} (F_{kI} F_{kJ} - \delta_{IJ})$	2nd Piola-Kirchhoff stress S_{IJ}	$\frac{dS_{IJ}}{dE_{IJ}}$
Engineering stress model	None	Engineering stress s_{ij}	$\frac{ds_{ij}}{d\varepsilon_{kl}}$

2.2.2 Stress update options

Table 2.2 lists the base classes available for implementing constitutive models. The table also describes any extra kinematic measures provided by the base class, and the stress and algorithmic tangent (Jacobian) tensors the user needs to implement to define the model. The subsequent subsections provide the mathematical theory used to connect the user-defined stress and tangent measure to the Cauchy stress, 1st Piola-Kirchhoff stress, and the associated tangent measures.

2.2.2.1 Cauchy stress material

This model converts the Cauchy stress and the algorithmic tangent to provide the 1st Piola Kirchhoff stress, where needed by the Lagrangian kernel system. The conversion formula are:

$$\sigma_{ij} = \frac{1}{J} P_{iK} F_{jK} \quad (2.31)$$

and

$$T_{ijkl} = \frac{d\sigma_{ij}}{d\Delta l_{kl}} = \frac{1}{J} T'_{iAmN} F_{jA} F_{lN} f_{mk} + f_{jk} \sigma_{il} - \sigma_{ij} f_{lk} \quad (2.32)$$

2.2.2.2 1st Piola-Kirchhoff stress material

This model converts the 1st Piola Kirchhoff stress and the algorithmic tangent to provide the Cauchy stress, where needed by the Lagrangian kernel system. The conversion formula are

$$P_{iJ} = J\sigma_{is}F_{Js}^{-1} \quad (2.33)$$

and

$$T'_{iJkL} = J\sigma_{im} (F_{Lk}^{-1}F_{Jm}^{-1} - F_{Jk}^{-1}F_{Lm}^{-1}) + JT_{isab}f_{ak}^{-1}F_{Lb}^{-1}F_{Js}^{-1} \quad (2.34)$$

2.2.2.3 2nd Piola-Kirchhoff stress material

The model maps from the 2nd Piola Kirchhoff stress to the 1st Piola Kirchhoff stress (and similarly maps the tangents). The formula above can then be used to convert to the Cauchy stress. The conversion formula are

$$P_{iJ} = F_{iK}S_{KJ} \quad (2.35)$$

and

$$T'_{iJkL} = \delta_{ik}S_{LJ} + F_{iT}T''_{TJMN}\frac{1}{2}(\delta_{ML}F_{kN} + F_{kM}\delta_{NL}) \quad (2.36)$$

2.2.2.4 Engineering (small) stress material

This model is a bit different from the other constitutive model options. It provides an interface for implementing a constitutive model using the traditional small deformation, engineering stress and strain measures that translates this natively small deformations constitutive model to provide a suitable response for a large deformation formulation, as implemented in the updated and total Lagrangian kernels. Specifically, the class provides a way to calculate the Cauchy stress given the small, engineering stress and to convert the algorithmic tangent provided by an engineering stress/strain model to a suitable large deformation tangent tensor.

The user then implements the small stress update s_{ij} and the associated derivative with respect to the engineering strain

$$\hat{T}_{ijkl} = \frac{ds_{ij}}{d\varepsilon_{kl}} \quad (2.37)$$

The class then converts these measures to the updated Cauchy stress. The conversion formula above can then convert to the first Piola-Kirchhoff stress for the total Lagrangian formulation.

The class converts the small strain model by integrating an objective rate of the Cauchy stress [7]. There are a wide variety of objective rates described in the literature and the implementation in the Tensor Mechanics module provides a general form in which different rates can be implemented. The current implementation provides two options: the Truesdell rate of the Cauchy stress and the Jaumann rate of the Cauchy stress. The implicit version of the commercial ABAQUS FEA code uses the Jaumann rate, so enabling this option allows users to compare results to that product.

The process starts by updating the small stress using the user-provided constitutive model, defined (typically) in terms of the mechanical strain tensor. However, as with all

constitutive models designed for use with the Lagrangian kernels, the user can define the stress update in terms of any kinematic measure.

Most objective rates take the form

$$\hat{\sigma}_{ij} = s_{ij} = \dot{\sigma}_{ij} - Q_{ik}\sigma_{kj} - \sigma_{ik}Q_{jk} + Q_{kk}\sigma_{ij} \quad (2.38)$$

where Q_{ik} is some kinematic measure and s_{ij} is the small stress, supplied by the constitutive model. This equation basically advects the stress using the kinematic tensor. The $\hat{\sigma}_{ij}$ suggests that multiple objective rates of the Cauchy stress are possible – i.e. there is no unique, universally accepted theory. The choice of the kinematic tensor Q_{ik} defines the particular objective rate, so long as the model returns the correct tangent matrix \hat{T}_{ijkl} .

The conversion process must solve this equation to find the updated Cauchy stress for an arbitrary kinematic measure Q . It turns out this update is linear and the solution for the updated Cauchy stress is

$$\sigma_{ij} = J_{ijmn}^{-1}(\sigma_{mn}^n + \Delta s_{mn}) \quad (2.39)$$

where Δs_{mn} is the increment in the small stress over some time step and

$$J_{ijmn} = (1 + \Delta Q_{kk})\delta_{im}\delta_{jn} - \Delta Q_{im}\delta_{jn} - \delta_{im}\Delta Q_{jn} \quad (2.40)$$

with $\Delta Q_{ij} = \Delta t Q_{ij}$, i.e. the increment in the kinematic tensor.

The algorithmic tangent for the Cauchy stress is then given by

$$T_{ijkl} = J_{ijmn}^{-1} \left(\hat{T}_{mnst} - \frac{\partial J_{mnst}}{\partial \Delta l_{kl}} \sigma_{st} \right) \quad (2.41)$$

where the $\frac{\partial J_{mnst}}{\partial \Delta l_{kl}}$ tensor is another characteristic of the objective rate. Rather than implement a 6th order tensor, the class instead implements a function giving the action of this tensor on the Cauchy stress, i.e.

$$U_{mnkl} = \frac{\partial J_{mnst}}{\partial \Delta l_{kl}} \sigma_{st} \quad (2.42)$$

The two tensors Q_{ij} and U_{ijkl} then fully-define a particular objective rate.

The current implementation provides two objective rate options.

The Truesdell rate of the Cauchy stress The Truesdell objective rate is defined by the kinematic tensor

$$Q_{ik} = l_{ik} \quad (2.43)$$

and the derivative tensor

$$U_{mnkl} = \delta_{kl}\sigma_{mn} - \delta_{mk}\sigma_{ln} - \delta_{nk}\sigma_{ml} \quad (2.44)$$

The Jaumann rate of the Cauchy stress The Jaumann objective rate is defined by

$$Q_{ik} = w_{ik} \quad (2.45)$$

with

$$w_{ik} = \frac{1}{2}(l_{ik} - l_{ki}) \quad (2.46)$$

and the associated derivative tensor

$$U_{mnkl} = \frac{1}{2}(\delta_{ml}\sigma_{kn} + \delta_{nl}\sigma_{mk} - \delta_{mk}\sigma_{ln} - \delta_{nk}\sigma_{ml}) \quad (2.47)$$

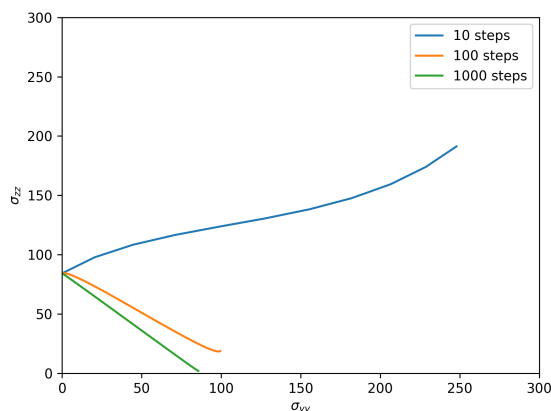


Figure 2.2: Plot of σ_{yy} and σ_{zz} in the cube as it rotates, for different numbers of time steps.

Problems with objective rates There are several well-known problems associated with integrating objective rates to provide large deformation constitutive models based on small strain theory [7, 8].

A typical use for objective rate integration is to provide a constitutive model for materials that only undergo relatively small stretches in a simulation that will require large rotations. For example, a user might want to simulate a cold work forming process for a metal part, where the material will not under large strains but will undergo large rotations. One challenge with objective rate integration as implemented here is that the rotation kinematics are integrated in time, rather than being applied directly from the simulation kinematics. This means that the models will accurately capture large rotations but only in the limit of zero time integration error. In theory then, the rotational kinematics are only correct for infinitesimal time steps.

A simple example illustrates the problem: a block of material is stretched, developing some stress, and then rotated 90°. A correct simulation of this process would first develop stress in the z -direction and then keep the magnitude of the stress constant as the block of material rotates. Figure 2.2 shows the zz and yy components of the Cauchy stress, as integrated for an elastic material with the Truesdell rate, during this process for different numbers of integration time steps during the rotational part of the deformation. For large numbers of time steps the simulation results are correct: the yy component of the Cauchy stress at the end of the simulation is equal to the initial zz stress and the zz component goes to zero as the block rotates. But for fewer steps the rotational process is not integrated exactly, leading to errors in the final stress tensor.

This 90° rotation without additional stretch is not a typical simulation. More often, a simulation would be deforming the material during the large rotations, which in turn requires a smaller time step to accurately resolve the material deformation itself. However, this example does illustrate one of the shortcomings of this particular implementation of objective integration. Note this is not a generic shortcoming of objective rates, other integration approaches can achieve exact rotational kinematics regardless of the time increment, including one of the options in the base tensor mechanics kernels [9].

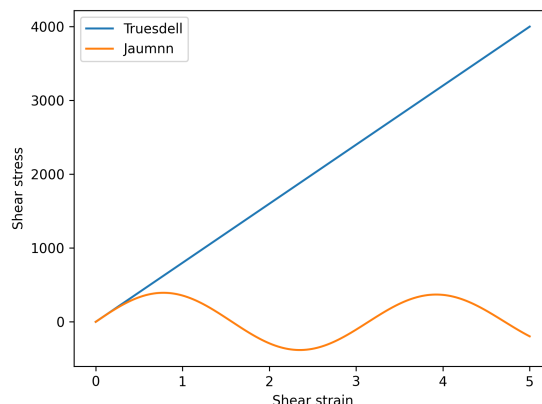


Figure 2.3: Shear stress/shear strain plot comparing the Truesdell and Jaumann rates for very large shear deformations.

Models may exhibit anomalous, unphysical behavior when subjected to large shear deformations. Figure 2.3 compares the results of shearing a block of material to very large shear strains using both the Truesdell and Jaumann rates. The shear stress/strain response for the Jaumann model oscillates, which is not a reasonable, physical response for the elastic material. The Truesdell rate, which is used by default by the new mechanics models, avoids this non-physical behavior.

2.3 \bar{F} stabilization

2.3.1 *The need for stabilization*

Standard 2D and 3D finite elements using a linear interpolation of the displacement field are not stable for incompressible and nearly incompressible deformation [10]. These standard elements include the triangle, tetrahedral, quadrilateral, and hexahedral elements commonly used in solving problems in the Tensor Mechanics module. Incompressible deformation is material deformation that does not change the local volume of the structure, for example linear elastic deformation as the Poisson's ratio approaches $\nu = 0.5$. Other common examples include problems representing elastic-plastic materials problems with widespread plasticity, as (traditional) plasticity occurs via shear, as well as many types of hyperelastic models representing soft polymers.

Under these conditions standard linear elements exhibit volumetric locking where the apparent, numerical stiffness of the element is much greater than the actual analytic stiffness of the structure. This locking leads to inaccurate results which do not improve with mesh refinement.

2.3.2 \bar{F} and \bar{B} Stabilization

There are two common methods used to stabilize linear elements and avoid volumetric locking: the \bar{F} [11] and \bar{B} [12]. Of these theories \bar{B} is older and was originally developed for

small deformation problems while \bar{F} was developed later and originally intended for large deformation problems. However, the \bar{F} method can be used for small deformation problems as well, \bar{B} can be extended to large deformation problems, and the \bar{F} method can be viewed as a subset of the \bar{B} theory. The following explains the two approaches in the context of small deformations.

\bar{F} alters the definition of the strain being fed into the constitutive models to produce the stress, subsequently used by the kernel to calculate the stress equilibrium residual. The theory modifies the strains so that the dilatational part of the strain at each quadrature point is set equal to the volume-average dilatation strain. Mathematically,

$$\bar{\varepsilon}_{ij} = \frac{1}{v} \int_v \varepsilon_{ij} dv \quad (2.48)$$

$$\varepsilon'_{ij} = \varepsilon_{ij} + \frac{1}{3} (\bar{\varepsilon}_{kk} - \varepsilon_{kk}) \delta_{ij} \quad (2.49)$$

where ε_{ij} is the strain calculated from the displacement gradient. This method then stabilizes the problem by replacing the linear-varying dilatational strain with a constant dilatational strain over each element. Notionally, in MOOSE the \bar{F} only alters the material model, though in fact it also changes the definition of the Jacobian (but not the residual) in the kernel.

\bar{B} makes this modification to the strain but then also modifies the definition of the residual, replacing the original small deformation stress equilibrium weak form

$$R^\alpha = \int_v s_{ij} \phi_{i,j}^\alpha dv \quad (2.50)$$

with a modified version

$$R^\alpha = \int_v s_{ij} \phi_{i,j}'^\alpha dv \quad (2.51)$$

where the method modifies the trial function gradient $\phi_{i,j}^\alpha$ in exactly the same way as the strains:

$$\bar{\phi}_{i,j}^\alpha = \frac{1}{v} \int_v \phi_{i,j}^\alpha dv \quad (2.52)$$

$$\phi_{i,j}'^\alpha = \phi_{i,j}^\alpha + \frac{1}{3} (\bar{\phi}_{k,k}^\alpha - \phi_{k,k}^\alpha) \delta_{ij}. \quad (2.53)$$

The \bar{B} modification results in a symmetric Jacobian (assuming that the original problem had a symmetric Jacobian). This is a significant advantage for codes taking advantage of the symmetry of the assembled Jacobian matrix. However, MOOSE does not take advantage of this symmetry and so the Lagrangian kernel system implements the \bar{F} method, as it is somewhat easier to derive and implement in the large deformation context.

2.3.3 Implementation of the \bar{F} method

The form of the stabilization depends on if the problem is using large or small displacement kinematic theory, For small displacements the strain calculator modifies the strains in the manner described in the previous subsection:

$$\bar{\varepsilon}_{ij} = \frac{1}{v} \int_v \varepsilon_{ij} dv \quad (2.54)$$

$$\varepsilon'_{ij} = \varepsilon_{ij} + \frac{1}{3} (\bar{\varepsilon}_{kk} - \varepsilon_{kk}) \delta_{ij} \quad (2.55)$$

For large displacements the strain calculator modifies the deformation gradient instead:

$$\bar{F}_{iJ} = \frac{1}{V} \int_V F_{iJ} dV \quad (2.56)$$

$$F'_{iJ} = \left(\frac{\det \bar{F}}{\det F} \right)^{1/3} F_{iJ}. \quad (2.57)$$

From here the stress update proceeds the same as with stabilization off, except the stress is now based on the modified strain value. The \bar{F} approach does not alter the weak form residual in the kernel. However, this change in the definition of the strain does affect the Jacobian calculated in the kernel.

2.3.4 Cook's membrane

Figure 2.4 shows *Cook's Membrane*, a classical problem for demonstrating volumetric locking and assessing stabilization techniques for overcoming it. When this problem is solved with a nearly incompressible material it induces locking in unstabilized, linear, Q4 quad elements. Figures 2.5 and 2.6 show the problem solved twice, first with small deformation kinematics and a linear elastic material defined by $E = 250$ and $\nu = 0.4999999$ and then again with large deformation kinematics and a Neo-Hookean material with $\lambda = 416666611.0991259$ and $\mu = 8300.33333888888926$. Each plot shows the displacement at the tip of the beam as a function of mesh refinement

These plots demonstrate:

1. The problem with locking in both large and small deformations for unstabilized, linear elements. The beam tip displacement is much smaller in the unstabilized problems compared to the true solution and the stabilized solutions (i.e. these elements are very stiff). Moreover, mesh refinement is not effective at resolving the issue.
2. The \bar{F} stabilization implemented in the Lagrangian kernel system effectively eliminates volumetric locking for both the updated and total Lagrangian formulations and for both small and large deformations. The stabilized solutions for each kernel type are identical, they demonstrate the proper, non-locking stiffness, and mesh refinement converges the problem to a stable solution.

2.4 Homogenization subsystem

The new mechanics system integrates into a subsystem for imposing cell-average strain or stress constraints over periodic simulation domains. Our previous report [1] describes the system in detail. This section summarizes the theory and its implementation into the new mechanics system.

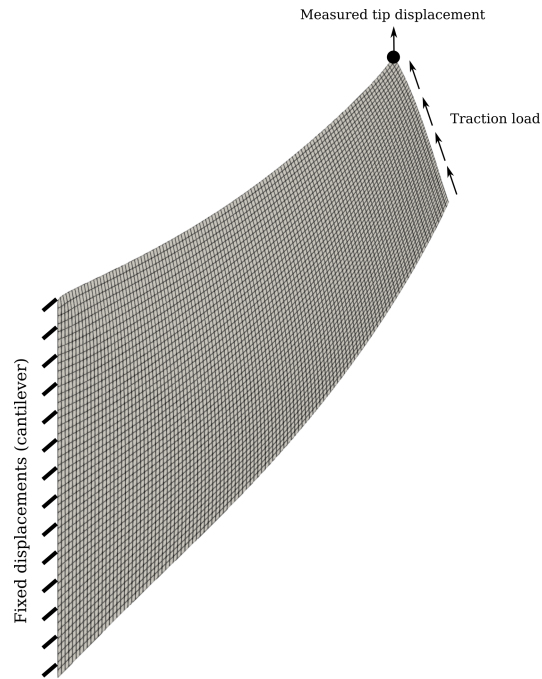


Figure 2.4: Cook's membrane: a reference problem for testing locking and stabilization strategies.

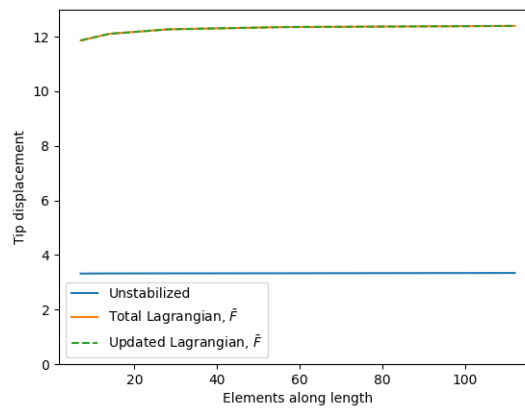


Figure 2.5: Demonstration of \bar{F} stabilization on a small deformation problem.

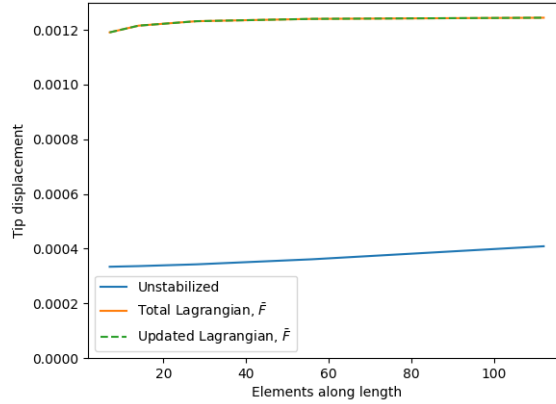


Figure 2.6: Demonstration of \bar{F} stabilization on a large deformation problem.

2.4.1 Mathematical description

Danielsson et al. [13] describes the theory underlying the homogenization system. The system aims to impose cell average stress or deformation constraints. For small deformations these constraints are of the type

$$\hat{s}_{ij} = \frac{1}{V} \int_v s_{ij} dV \quad (2.58)$$

or

$$\hat{\varepsilon}_{ij} = \frac{1}{V} \int_V \varepsilon_{ij} dV \quad (2.59)$$

and for large deformations the constraint options are:

$$\hat{P}_{iJ} = \frac{1}{V} \int_V P_{iJ} dV \quad (2.60)$$

or

$$\hat{F}_{iJ} - \delta_{iJ} = \frac{1}{V} \int_V (F_{iJ} - \delta_{iJ}) dV. \quad (2.61)$$

The “hat” quantities are targets set by the user with a MOOSE function. These targets control the cell-average stress or strain. For large deformations the system imposes the constraints on the 1st Piola-Kirchhoff stress or the displacement gradient. For small deformations the system constrains the small stress or the small strain.

Components can be mixed and matched – the system can impose deformation constraints in one tensor direction and stress constrains in the others. The number of available constraints varies by the problem dimension and the kinematic theory.

2.4.2 Example simulation

Figure 2.7 shows a simple, but representative, sample simulation. It consists of a square, 2D simulation cell subdivided into four subdomain. The material model in each subdomain is a St. Venant-Kirchhoff model, but the material properties in each subdomain are different

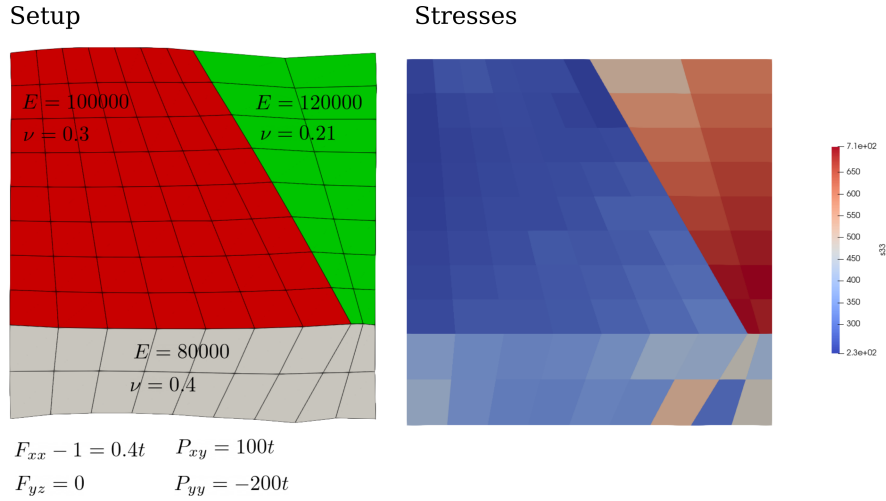


Figure 2.7: Simple, 2D large deformation homogenization cell problem.

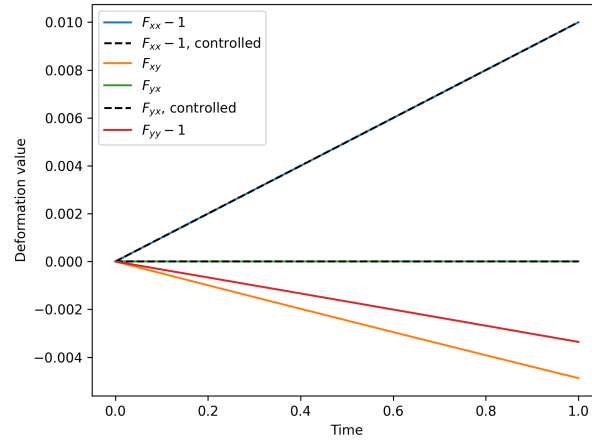


Figure 2.8: Results showing that the controlled deformation values match their targets.

as indicated on the figure. The problem imposes a combination of deformation gradient and 1st Piola Kirchhoff constraints on the simulation cell.

Figures 2.8 and 2.9 illustrate that the cell average stress and strain match the constraints for the controlled components of the 1st Piola-Kirchhoff and deformation gradient tensor. The uncontrolled components can vary to maintain equilibrium in the body. Because each subdomain has different material properties the microdisplacements are not homogeneous, producing an inhomogeneous stress state in the simulation cell.

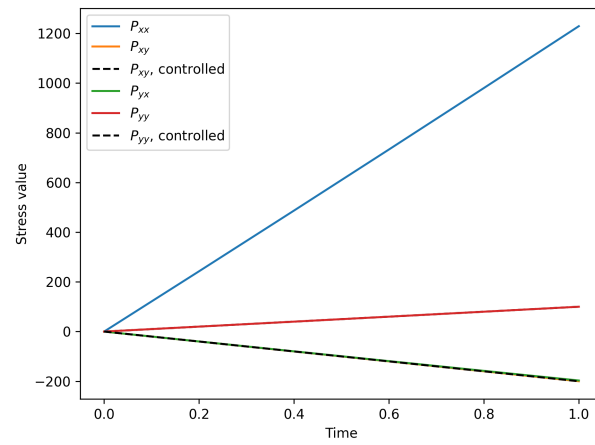


Figure 2.9: Results showing that the controlled stress values match their targets.

3 Integrating the Interface-Cohesive System into MOOSE

This chapter describes the integration of an interface-cohesive modeling framework, long under developed at ANL, into mainline MOOSE. As with the solids system, this ended up requiring substantial new, additional work both to integrate the existing material system and fundamental changes in the underlying kinematic formulation to improve the performance of the final models. The new implementation of the cohesive zone modeling framework in MOOSE has the following improvements compared its predecessor:

- it is modular, thus allowing for an easy implementation of additional traction-separation constitutive models,
- it supports both small strain and and large strain formulations,
- it allows using total and total and incremental constitutive traction separation laws for all kinematics formulations,
- it provides the exact Jacobian,
- the new `CohesiveZoneMaster` action simplifies the output syntax (same as the current `TensorMechanicsMaster` action), and
- the total Lagrangian formulation is objective, and does not suffer from numerical integration error (i.e. for an elastic material it does not dissipates energy even for very large deformations)

The implemented framework uses four kinds of objects to build the cohesive zone modeling system:

1. a displacement jump calculator,
2. a global traction calculator,
3. one, two, or three cohesive interface kernels (one for each problem dimension), and
4. a traction separation constitutive model.

Displacement jump calculators are responsible for providing the displacement jump in interface coordinates, $\llbracket \hat{u} \rrbracket$, and are implemented inheriting from *CZMComputeDisplacementJumpBase*. The global traction calculator objects compute the traction used to impose equilibrium, t , and its derivatives with respect to kinematic variables, $\partial t_i / \partial \llbracket u \rrbracket_j$ and $\partial t_i / \partial F_{jk}$. These objects inherit from *CZMComputeGlobalTractionBase*. Each cohesive interface kernels computes the residual R_i^\pm associated to the displacement component i and adds it to the equation system. This type of kernels also assembles the Jacobian, $\frac{\partial R_i^\pm}{\partial u_j^\pm}$, and should be implemented inheriting from *CZMInterfaceKernelBase*. The *CohesiveZoneMaster* action simplifies the input file syntax by adding the proper *CZMComputeDisplacementJump*, *CZMComputeGlobalTraction*, and *CZMInterfaceKernels*. The user is only responsible for adding the proper *CZMComputeLocalTraction* material in the input file material section. The schematic in Figure 3.1 depicts the flow of information between the different objects and highlights the objects automatically added by the action.

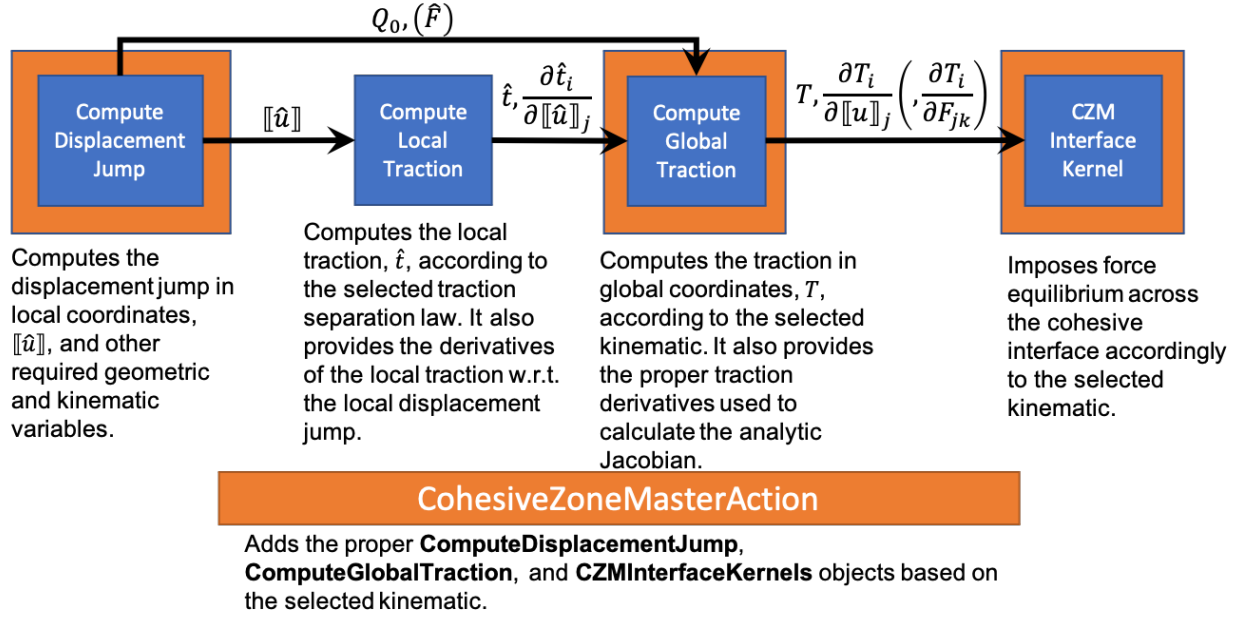


Figure 3.1: Schematic depicting the objects required for cohesive zone modeling in *MOOSE*, and the objects automatically added by the *CohesizeZoneMaster* action

Traction separation constitutive models are implemented using a *small deformation* formulation. The system applies a kinematic update to this base model to accomodate large deformation problems. The framework provides two base classes for implementing traction separation laws:

1. *CZMComputeLocalTractionTotalBase*, and
2. *lightgrayCZMComputeLocalTractionIncrementalBase*.

The total formulation assumes the interface traction is only function of the total displacement jump, i.e. $\hat{t} = f([[\hat{u}]])$, and can be used to implement traction separation laws without damage, such as the model described in [14] and [15]. The incremental formulation assumes the traction rate depends upon the displacement jump increment and the values of one or more internal variables, i.e. $\Delta \hat{t} = f([[\Delta \hat{u}]], h)$. Thus, the incremental formulation should be used for path dependent traction separation laws, such as the numerical implementation of the PPR model described in [16].

When implementing a new traction separation law the user should override the *computeInterfaceTractionAndDerivatives*. For the total formulation, the user provides the interface traction and its derivatives with respect to the interface displacement jump, and saves them into *_interface_traction[_qp]* and *_dinterface_traction.djump[_qp]* variables, respectively. For the incremental formulation, the user provides the interface traction increment and its derivatives w.r.t. to the interface displacement jump, and saves them into *_interface_traction_inc[_qp]* and *_dinterface_traction.djump[_qp]* variables, respectively. Furthermore, the total interface traction is automatically calculated as $\hat{t} = \Delta \hat{t} + \hat{t}_{old}$.

3.1 Large deformation Cohesive zone modeling approach

3.1.1 The interface deformation gradient

The *MOOSE* framework does not support standard interface elements, therefore it is not possible to directly calculate the deformation gradient on the interface midplane, F . Knowing F is critical because the interface coordinate system is attached to the interface midplane and most, if not all, traction separation laws are written in terms of the normal and tangential displacement jump components. Furthermore knowing F also allows the interface system to push forward or pull backward the traction between two configurations. Figure 3.2 is a schematic depicting the two surfaces defining the cohesive interfaces and the interface midplane, their deformation gradient, and the interface coordinate system in the deformed and undeformed configurations.

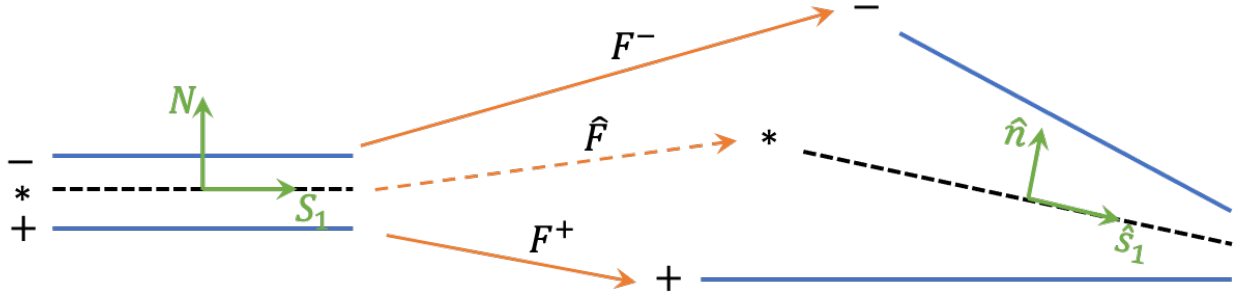


Figure 3.2: Schematic showing the bulk material surfaces (blue lines) and the interface midplane (dashed black lines) in the deformed and undeformed configuration and the associated deformation gradients. The figure also shows how the interface coordinate system evolves according to the deformation of interface midplane (green arrows) Note: in the undeformed configuration all surfaces coincide, however they are shown separated for clarity.

The *MOOSE* framework provides additional quadrature points on the bulk element surfaces, allowing the calculation of the deformation gradient of the cohesive surfaces. Therefore we assume the interface midplane deformation gradient, F , to be defined as the average deformation gradient of the initially coincident quadrature points:

$$F = \frac{1}{2} (F^+ + F^-) \quad (3.1)$$

where the superscripts $+$ and $-$ identify the two initially coincident surfaces.

The *InterfaceKernel* system available in *MOOSE* assumes elements are connected, thus implying the two sides of the interface always have the same area. In the deformed configuration the cohesive surfaces might have different areas, however in the undeformed configuration all surfaces are coincident. Therefore we elected to design the large deformation cohesive zone model using a *total Lagrangian* formulation, which impose equilibrium in the undeformed configuration.

In this manuscript, and also in the implementation, we define the normal cohesive traction to be positive when the normal displacement jump is positive. Since the *MOOSE*

InterfaceKernel provides the outward normal of the + surface, we define the displacement jump as

$$[[u]] = u^- - u^+ \quad (3.2)$$

Equation 3.2 provides a positive value for the normal jump when the cohesive surfaces are opening. In the same equation we also introduced the jump operator which is defined as $[[x]] = x^- - x^+$ where x can be a scalar, a vector, or a tensor.

3.1.2 Interface Coordinate Systems

The interface coordinate system is attached to the interface midplane and rotates with the material deformation and rigid body rotations. Displacements are always expressed in the global coordinate system, therefore a map transforming between the two coordinate systems is required to compute the normal and tangent component of the displacement jump. Under the small strain assumption, the interface rotation and area change are ignored. Hence, the mapping is the rotation matrix transforming from the interface coordinate system to the global coordinate system in the undeformed configuration. Assume \hat{N} , \hat{S}_1 and \hat{S}_2 are the normal and tangent unit vectors defining the interface coordinate system, and Q_0 is the rotation matrix transforming the normal and tangential components from interface coordinate system to the global coordinate system in the undeformed configuration:

$$\begin{bmatrix} N \\ S_1 \\ S_2 \end{bmatrix} = Q_0 \begin{bmatrix} \hat{N} \\ \hat{S}_1 \\ \hat{S}_2 \end{bmatrix} \quad (3.3)$$

Where N , S_1 and S_2 represent the interface normal and tangential vectors in global coordinates, in the undeformed configuration.

Finite strain kinematics must account for interface rotation caused by the applied loads and rigid body motion. We define R as the rotation transforming the interface normal and tangential vector, from the undeformed to the deformed configuration: During deformation N , S_1 and S_2 rotates according to the rotation R such that

$$\begin{bmatrix} n \\ s_1 \\ s_2 \end{bmatrix} = R \begin{bmatrix} N \\ S_1 \\ S_2 \end{bmatrix} \quad (3.4)$$

Where n , s_1 and s_2 represent the interface normal and tangential vectors in the global coordinates in the deformed configuration. Note that for the small strain assumption R is the identity matrix.

By using Equations 3.3 and 3.4 one can relate the interface normal and tangential vectors in the deformed configuration in global coordinates with the interface local coordinate system as

$$\begin{bmatrix} n \\ s_1 \\ s_2 \end{bmatrix} = Q \begin{bmatrix} \hat{N} \\ \hat{S}_1 \\ \hat{S}_2 \end{bmatrix} \quad (3.5)$$

with Q defined as:

$$Q = RQ_0 \quad (3.6)$$

The rotation matrix R is calculated computing the right polar decomposition of the interface deformation gradient, i.e. assuming $F = RU$ Figure 3.3 is a visual representation of the coordinate transformation process described above.

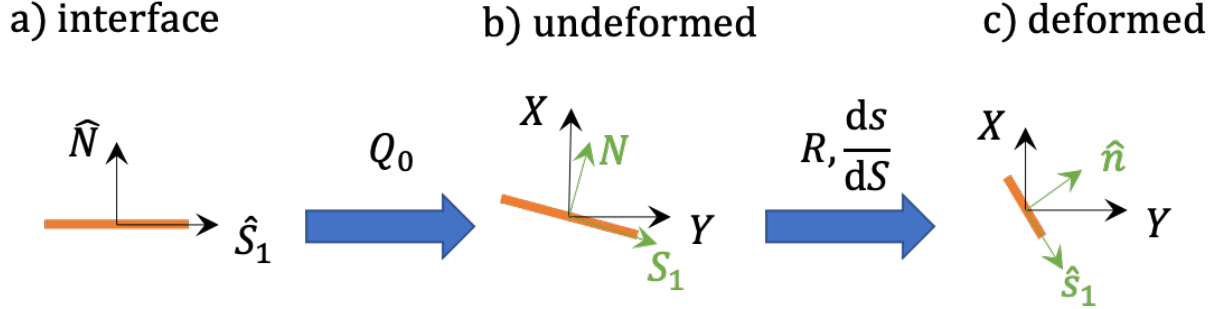


Figure 3.3: a) Interface coordinate system, b) interface coordinate system in global coordinates for the undeformed configuration, and c) the interface coordinate system in the deformed configuration. Q_0 is rotation matrix transforming from the interface coordinate system to the global coordinate system in the undeformed configuration. The rotation matrix R rotates a vector from the deformed to the undeformed configuration. The variables a and A are the interface areas in the deformed and undeformed configuration, respectively.

To compute the analytic Jacobian, we will need the partial derivative of the rotation matrix Q with respect to the deformation gradient. Since $Q = RQ_0$ and Q_0 is constant then

$$\frac{\partial Q_{ij}}{\partial F_{pq}} = \frac{\partial R_{ik}}{\partial F_{pq}} Q_{0,kj} \quad (3.7)$$

Chen and Wheeler obtained the following analytic expression[17]:

$$\frac{\partial R_{kl}}{\partial F_{pq}} = \frac{1}{\det(Y)} R_{kp} (Y_{pq} R_{mq} Y_{nl} - Y_{pn} R_{mq} Y_{ql}) \quad (3.8)$$

where $Y = \text{trace}(U)I - U$

3.1.3 Total Lagrangian Residual

Linear momentum equilibrium requires the cohesive infinitesimal forces to be equal and opposite:

$$df^+ + df^- = 0 \quad (3.9)$$

In the undeformed configuration angular momentum is automatically satisfied because surfaces are coincident. The infinitesimal force is defined as the traction multiplied by the infinitesimal area dS . Substituting the definition of infinitesimal forces in Equation 3.9 one obtains

$$T_i^- dS^- + T_i^+ dS^+ = 0 \quad (3.10)$$

where $i = x, y, z$ is the index associated to a particular Cartesian component. Since cohesive surfaces area coincident, their infinitesimal areas are identical and therefore cohesive traction

are equal and opposite ($T^+ = T^- = T$). Realizing displacements are the work conjugate of forces, the weak form of Equation 3.10 is

$$T_i \psi_i^{z,-} dS = T_i \psi_i^{z,+} dS \quad (3.11)$$

where $\psi(z, -)$ and $\psi(z, +)$ are the test function of the $+$ and $-$ surfaces, and z is test function index. Therefore the residuals for the $+$ and $-$ surfaces, for the test function z are

$$R_i^+(z) = -T_i \psi_i^{+,z} \quad (3.12)$$

$$R_i^-(z) = T_i \psi_i^{-,z} \quad (3.13)$$

In Equations 3.12 and 3.13 we dropped the dS dependency to highlight the residual implemented in the *CZMInterfaceKernelTotalLagrangian* object.

3.1.3.1 Total Lagrangian Jacobian

To achieve quadratic convergence the Jacobian must be exact. The displacements on both sides of the interface are the only variables involved in calculating the residual. In the Galerkin method variables are discretized as follows

$$u \approx \sum_z \phi^z u^z \quad (3.14)$$

where z is the test function index, ϕ^z is the test function, and u^z is the test function coefficient. Hence the analytic Jacobian with respect to the shape function coefficients is:

$$\frac{\partial R_i^+}{\partial u_j^{+,w}} \psi_i^{+,z} = -\frac{\partial T_i}{\partial u_j^{+,w}} \psi_i^{+,z} \quad (3.15)$$

$$\frac{\partial R_i^+}{\partial u_j^{-,w}} \psi_i^{+,z} = -\frac{\partial T_i}{\partial u_j^{-,w}} \psi_i^{+,z} \quad (3.16)$$

$$\frac{\partial R_i^-}{\partial u_j^{+,w}} \psi_i^{-,z} = \frac{\partial T_i}{\partial u_j^{+,w}} \psi_i^{-,z} \quad (3.17)$$

$$\frac{\partial R_i^-}{\partial u_j^{-,w}} \psi_i^{-,z} = \frac{\partial T_i}{\partial u_j^{-,w}} \psi_i^{-,z} \quad (3.18)$$

The first Piola-Kirchoff traction depends on the displacement jump $\llbracket u \rrbracket$ and other kinematic variables derived from the interface deformation gradient F . Hence the partial derivatives of the traction with respect to the discrete displacements can be expanded using the chain rule:

$$\frac{\partial T_i}{\partial u_j^+} = \frac{\partial T_i}{\partial \llbracket u \rrbracket_p} \frac{\partial \llbracket u \rrbracket_p}{\partial u_j^{+,w}} + \frac{\partial T_i}{\partial \hat{F}_{pq}} \frac{\partial \hat{F}_{pq}}{\partial u_j^{+,w}} \quad (3.19)$$

$$\frac{\partial T_i}{\partial u_j^{-,w}} = \frac{\partial T_i}{\partial \llbracket u \rrbracket_p} \frac{\partial \llbracket u \rrbracket_p}{\partial u_j^{-,w}} + \frac{\partial T_i}{\partial \hat{F}_{pq}} \frac{\partial \hat{F}_{pq}}{\partial u_j^{-,w}} \quad (3.20)$$

Using Equation 3.2 and 3.14 the derivatives of the jump with respect to the discrete displacements are:

$$\frac{\partial \llbracket u \rrbracket_p}{\partial u_j^{+,w}} = -\delta_{ij} \phi_j^{+,w} \quad (3.21)$$

$$\frac{\partial \llbracket u \rrbracket_p}{\partial u_j^{-,w}} = \delta_{ij} \phi_j^{-,w} \quad (3.22)$$

The deformation gradient itself is defined as:

$$F_{ij} = \delta_{ij} + \nabla u_{ij} = I + \begin{bmatrix} \frac{\partial u_1}{\partial X_1} & \frac{\partial u_1}{\partial X_2} & \frac{\partial u_1}{\partial X_3} \\ \frac{\partial u_2}{\partial X_1} & \frac{\partial u_2}{\partial X_2} & \frac{\partial u_2}{\partial X_3} \\ \frac{\partial u_3}{\partial X_1} & \frac{\partial u_3}{\partial X_2} & \frac{\partial u_3}{\partial X_3} \end{bmatrix} \quad (3.23)$$

where X are coordinates in the reference configuration and I is the identity matrix. Recalling that $\partial u_i / \partial X_j = \sum_w \nabla \phi_{ij}^w u_i^w$, the derivative of Equation 3.1 with respect to the discrete displacements are:

$$\frac{\partial F_{pq}}{\partial u_j^{+,w}} = \frac{1}{2} \begin{bmatrix} \begin{bmatrix} \nabla \phi_{1,1}^{+,w} & \nabla \phi_{1,2}^{+,w} & \nabla \phi_{1,3}^{+,w} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{j=1} \\ \begin{bmatrix} 0 & 0 & 0 \\ \nabla \phi_{2,1}^{+,w} & \nabla \phi_{2,2}^{+,w} & \nabla \phi_{2,3}^{+,w} \\ 0 & 0 & 0 \end{bmatrix}_{j=2} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \nabla \phi_{3,1}^{+,w} & \nabla \phi_{3,2}^{+,w} & \nabla \phi_{3,3}^{+,w} \end{bmatrix}_{j=3} \end{bmatrix} \quad (3.24)$$

$$\frac{\partial F_{ij}}{\partial u_k^{-,w}} = \frac{1}{2} \begin{bmatrix} \begin{bmatrix} \nabla \phi_{1,1}^{-,w} & \nabla \phi_{1,2}^{-,w} & \nabla \phi_{1,3}^{-,w} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{j=1} \\ \begin{bmatrix} 0 & 0 & 0 \\ \nabla \phi_{2,1}^{-,w} & \nabla \phi_{2,2}^{-,w} & \nabla \phi_{2,3}^{-,w} \\ 0 & 0 & 0 \end{bmatrix}_{j=2} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \nabla \phi_{3,1}^{-,w} & \nabla \phi_{3,2}^{-,w} & \nabla \phi_{3,3}^{-,w} \end{bmatrix}_{j=3} \end{bmatrix} \quad (3.25)$$

The *CZMInterfaceKernelTotalLagrangian* object uses Equations 3.15-3.20 to calculate the analytic Jacobian, and computes partial derivatives of $\llbracket u \rrbracket$ and \hat{F} . The partial derivative of T are provided by the *CZMComputeGlobalTractionTotalLagrangian* object.

Using a *Total Lagrangian* approach requires computing the first Piola-Kirchoff traction, T , which is defined as the ratio between the infinitesimal force df and the undeformed area, dA . The Cauchy traction t instead is the ratio between the infinitesimal force df and the current infinitesimal area da . Using force equilibrium and the above definitions we can write

$$T = \frac{da}{dA} t \quad (3.26)$$

Equation 3.26 also shows the Cauchy and first Piola-Kirchoff traction are collinear, and scaled by the ratio between the current and undeformed areas.

3.2 The cohesive zone material system

3.2.1 Cohesive zone constitutive models

The system assumes traction separation constitutive models calculate the interface local traction or its finite increment as function of the local displacement jump or its finite increment, i.e $\hat{t} = f(\llbracket \hat{u} \rrbracket)$ or $\Delta \hat{t} = f(\llbracket \Delta \hat{u} \rrbracket)$. Path dependent traction separation models can be implemented overriding the *CZMComputeLocalTractionTotalBase* object, while path independent models typically use the *CZMComputeLocalTractionIncrementalBase* object. Both types of materials can be used for the small strain or large strain simulations.

The local displacement jump or its increment are provided using the following order $(\hat{N}, \hat{S}_1, \hat{S}_2)$. The same convention is used for the traction vectors.

In this objects the user should also provide the analytic Jacobian of the local traction, or its increment, w.r.t. the total displacement jump. Notice that by construction $\partial \hat{t} / \partial \llbracket u \rrbracket = \partial \Delta \hat{t} / \partial \llbracket \Delta u \rrbracket$) The convention used for the Jacobian assumes a row major order:

$$\frac{\partial \hat{t}}{\partial \llbracket u \rrbracket} = \begin{bmatrix} \frac{\partial \hat{t}_N}{\partial \llbracket u \rrbracket_{\hat{N}}} & \frac{\partial \hat{t}_N}{\partial \llbracket u \rrbracket_{\hat{S}_1}} & \frac{\partial \hat{t}_N}{\partial \llbracket u \rrbracket_{\hat{S}_2}} \\ \frac{\partial \hat{t}_{\hat{S}_1}}{\partial \llbracket u \rrbracket_{\hat{N}}} & \frac{\partial \hat{t}_{\hat{S}_1}}{\partial \llbracket u \rrbracket_{\hat{S}_1}} & \frac{\partial \hat{t}_{\hat{S}_1}}{\partial \llbracket u \rrbracket_{\hat{S}_2}} \\ \frac{\partial \hat{t}_{\hat{S}_2}}{\partial \llbracket u \rrbracket_{\hat{N}}} & \frac{\partial \hat{t}_{\hat{S}_2}}{\partial \llbracket u \rrbracket_{\hat{S}_1}} & \frac{\partial \hat{t}_{\hat{S}_2}}{\partial \llbracket u \rrbracket_{\hat{S}_2}} \end{bmatrix} \quad (3.27)$$

3.2.2 The displacement jump provider

The new cohesive zone system provides two objects: *ComputeDisplacementJumpSmallStrain*, and *ComputeDisplacementJumpFiniteStrain*.

The first object computes the interface displacement jump as:

$$\llbracket \hat{u} \rrbracket = Q_0^T \llbracket u \rrbracket. \quad (3.28)$$

The *ComputeDisplacementJumpFiniteStrain* object is also responsible for computing the interface deformation gradient F and its polar decomposition because it needs the rotation matrix R to properly compute the displacement jump in interface coordinates:

$$\llbracket \hat{u} \rrbracket = Q^T \llbracket u \rrbracket \quad (3.29)$$

where Q is the rotation matrix defined by Equation 3.6.

3.3 Integral PK1 traction formulation

The first Piola-Kirchoff traction, T and the Cauchy traction, t are related by Equation 3.26. The material models provides the interface traction in the deformed configuration and can be converted to the Cauchy traction t by the rotation matrix Q as follow:

$$t = Q\hat{t} \quad (3.30)$$

Substituting Equation 3.30 in 3.26 we can write:

$$T = \frac{da}{dA} Q\hat{t} \quad (3.31)$$

which relates the first Piola-Kirchoff traction to the interface traction directly. By manipulating Nanson's formula it can be show that the area ratio is:

$$\frac{da}{dA} = \det(F) \|F_{ij}^{-T} N_j\|_2 \quad (3.32)$$

3.3.1 Derivative of T with respect to the discrete displacements

As shown in section 3.1.3.1, computing the analytic Jacobian requires computing the derivative of the traction with respect to the discrete displacements. Equations 3.19 and 3.20 have two unknown terms: $\partial T_i / \partial [u]_p$ and $\partial T_i / \partial F_{pq}$. Using the chain rule and assuming only the interface traction directly depends from the $[u]$ the first term can be expanded as:

$$\frac{\partial T_i}{\partial [u]_p} = \frac{da}{dA} Q_{ij} \frac{\partial \hat{t}_j}{\partial [\hat{u}]_w} \frac{\partial [\hat{u}]_w}{\partial [u]_p} \quad (3.33)$$

The derivative of the interface traction with respect to the interface displacement jump, i.e $\partial \hat{t}_j / \partial [\hat{u}]_w$ is provided by the constitutive model. The second partial derivative instead is computed using the definition of interface displacement jump (Equations 3.28 or 3.29). Therefore

$$\frac{\partial [\hat{u}]_w}{\partial [u]_p} = Q_{wp}^T \quad (3.34)$$

Equation 3.34 is valid for the both the total and incremental formulations, because $\partial [\hat{u}]_{old} / \partial [u] = 0$. Substituting Equation 3.34 in Equation 3.33 we can write:

$$\frac{\partial T_i}{\partial [u]_p} = \frac{da}{dA} Q_{ij} \frac{\partial \hat{t}_i}{\partial [\hat{u}]_w} Q_{wp}^T \quad (3.35)$$

which is the implemented equation.

Using the chain rule and Equation 3.31, the partial derivative of T w.r.t. F reads:

$$\frac{\partial T_i}{\partial F_{pq}} = \left(\frac{\partial \frac{da}{dA}}{\partial F_{pq}} Q_{ij} + \frac{da}{dA} \frac{\partial Q_{ij}}{\partial F_{pq}} \right) \hat{t}_j + \frac{da}{dA} Q_{ij} \frac{\partial \hat{t}_j}{\partial F_{pq}} \quad (3.36)$$

The derivative of the area ratio with respect to the interface deformation gradient is:

$$\frac{\partial \frac{da}{dA}}{\partial F_{pq}} = \det(F) \left(F_{pq}^{-T} \|F_{ij}^{-T} N_j\|_2 - \frac{F_{ij}^{-T} N_j}{\|F_{ij}^{-T} N_j\|_2} F_{ip}^{-T} F_{qr}^{-T} N_r \right) \quad (3.37)$$

and the partial derivative of Q w.r.t. F is computed using Equation 3.7. Assuming the interface traction is only function of the interface displacement jump, then its derivative w.r.t F is:

$$\frac{\partial \hat{t}_j}{\partial F_{pq}} = \frac{\partial \hat{t}_j}{\partial \llbracket \hat{u} \rrbracket_v} \frac{\partial \llbracket \hat{u} \rrbracket_v}{\partial F_{pq}} \quad (3.38)$$

By using equation 3.29 it can be shown that:

$$\frac{\partial \hat{t}_j}{\partial F_{pq}} = \frac{\partial \hat{t}_j}{\partial \llbracket \hat{u} \rrbracket_v} \frac{\partial Q_{vw}^T}{\partial F_{pq}} \llbracket u \rrbracket_w \quad (3.39)$$

Substituting Eq. 3.39 in Eq. 3.36 and rearranging provides:

$$\frac{\partial T_i}{\partial F_{pq}} = \frac{\partial \frac{da}{dA}}{\partial F_{pq}} Q_{ij} \hat{t}_j + \frac{da}{dA} \left(\frac{\partial Q_{ij}}{\partial F_{pq}} \hat{t}_j + Q_{ij} \frac{\partial \hat{t}_j}{\partial \llbracket \hat{u} \rrbracket_v} \frac{\partial Q_{vw}^T}{\partial F_{pq}} \llbracket u \rrbracket_w \right) \quad (3.40)$$

The *CZMGlobalTractionCalcualtorTotalLagrangian* material computes the partial derivatives of the traction using Equations 3.35, 3.37, 3.7, 3.8, and 3.40. Such derivatives are used by the *CZMInterfaceKernleTolalLagrangian* to compute the exact Jacobian.

3.3.1.1 Testing

Figure 3.4 depict the resulting Cauchy and first Piola-Kirchoff traction components for a stretch plus rotate test obtained using the large-deformation, integral, cohesive zone model formulation for two cases:

1. using the total material formulation and and elastic traction separation law, i.e. $\hat{t} = K \llbracket \hat{u} \rrbracket$, and
2. using the incremental material formulation and and elastic traction separation law, i.e. $\Delta \hat{t} = K \llbracket \Delta \hat{u} \rrbracket$.

Lines and squares represents the first and second case, respectively. As expected the two model produce the same results, the first Piola-Kirchoff traction increases linearly with stretch, the Cacuchy traction accounts for the area changes, and the Cartesian components of both traction exactly follow the rotation. The green, horizontal line represents the normal magnitude of the Cauchy traction, which remains constant during rotation.

3.3.2 The small strain formulation

The small strain, formulation is identical to total Lagrangian formulation except the deformation gradient F is replaced with the identity matrix. Hence, all partial derivatives with respect to F are zero, the area ratio is always 1, and the rotation matrix Q is equal to Q_0 . To improve numerical efficiency for the small strain approach we implement three additional objects: *CZMDisplacementJumpSmallStrain*, *CZMComputeGlobalTractionSmallStrain*, and the *CZMInterfaceKernelSmallStrain* which avoid explicitly calculating these zero derivatives.

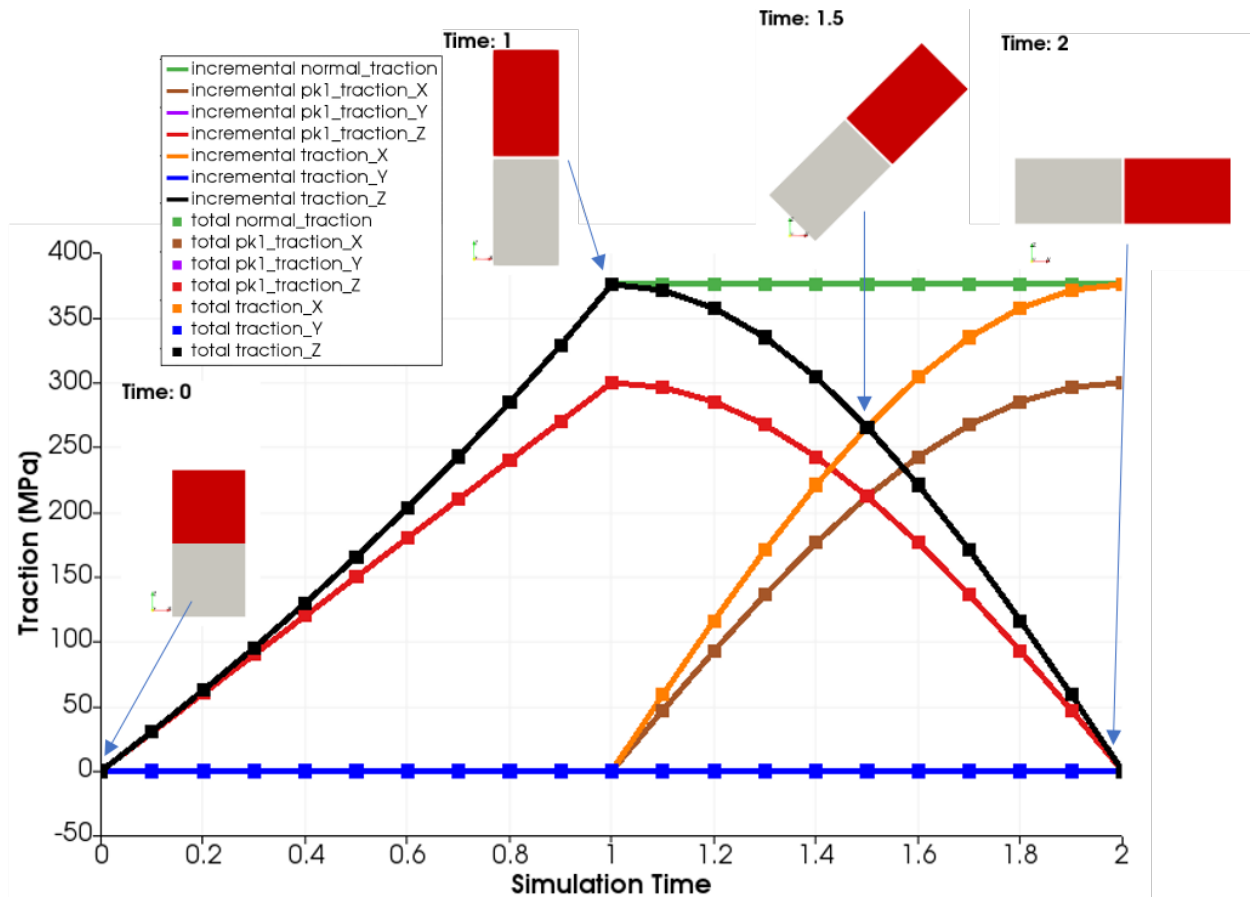


Figure 3.4: Stretch (simulation time 0-1) plus rotation (90° around the y-axis, simulation time 1-2) test using a linear traction separation law implemented using the total and incremental formulation.

4 Full-scale CPFEM Simulations of Cyclic Load

One goal of the work described here is to provide an ability for MOOSE to run full field simulations of creep deformation and failure in components ore realistic models of test specimens. To demonstrate this capability, crystal plasticity (CP) simulations were performed on full-scale microstructures, under stress controlled loading conditions to model creep behavior. Input microstructures with varying aspect ratios and geometries, with and without free surfaces were used to study the effect of free surfaces on creep performance. The literature lacks conclusive evidence to establish the effects of free surfaces on creep performance. The present work addresses the above problem using CPFEM simulations on Grade 91 steel.

4.1 Microstructure and loading conditions

An example input microstructure is shown below in Figure 4.1 (a). This representative cylindrical microstructure has a 1000 grains and approximately 1.5 M elements. The maximum simulation stress used was 100 MPa, which was ramped up to in the first 0.1 hrs, and then held constant (creep load) for a maximum of $10^6 hrs$, as shown in Figure 4.1 (b). The microstructure was loaded along the top face of the z-axis, with the bottom face was fixed. The rest of the surfaces were free of constraints. The microstructure shown in Figure 4.1 (a) has an aspect ratio of ~ 6 ; other microstructures with aspect ratios varying from 1 to 10 were also considered in this study. Also, cubic microstructures were simulated (with and without free surfaces) to compare the creep response with the cylindrical microstructure. The grain size was maintained consistent at $60 \mu m$ (consistent with experimental observations for Grade 91 steel), and the element sizes were maintained constant across the different simulated microstructures. The constitutive model used for these simulations is the creep model developed earlier in the group[6], where creep damage is modeled by means of grain boundary cavitation mechanism. Finally, the NEML material model framework [18] was used in conjunction with the MOOSE finite element solver [19] for the crystal plasticity simulations.

The large scale of the simulation setup required the implementation of an updated meshing algorithm for better performance of the finite element simulations. A comparison of the meshes resulting from the old and new methods are shown in Figure 4.2. Previously, Gmsh [20] was used to mesh the microstructure, but as is seen in Figure 4.2 (a), Gmsh is unable to handle efficiently the curvature of the cylindrical microstructure. Coreform Trelis [21] was instead used to produce higher quality meshes as shown in Figure 4.2 (b). The resulting mesh from Trelis provides more consistently shaped elements, which subsequently helps finite element simulation convergence.

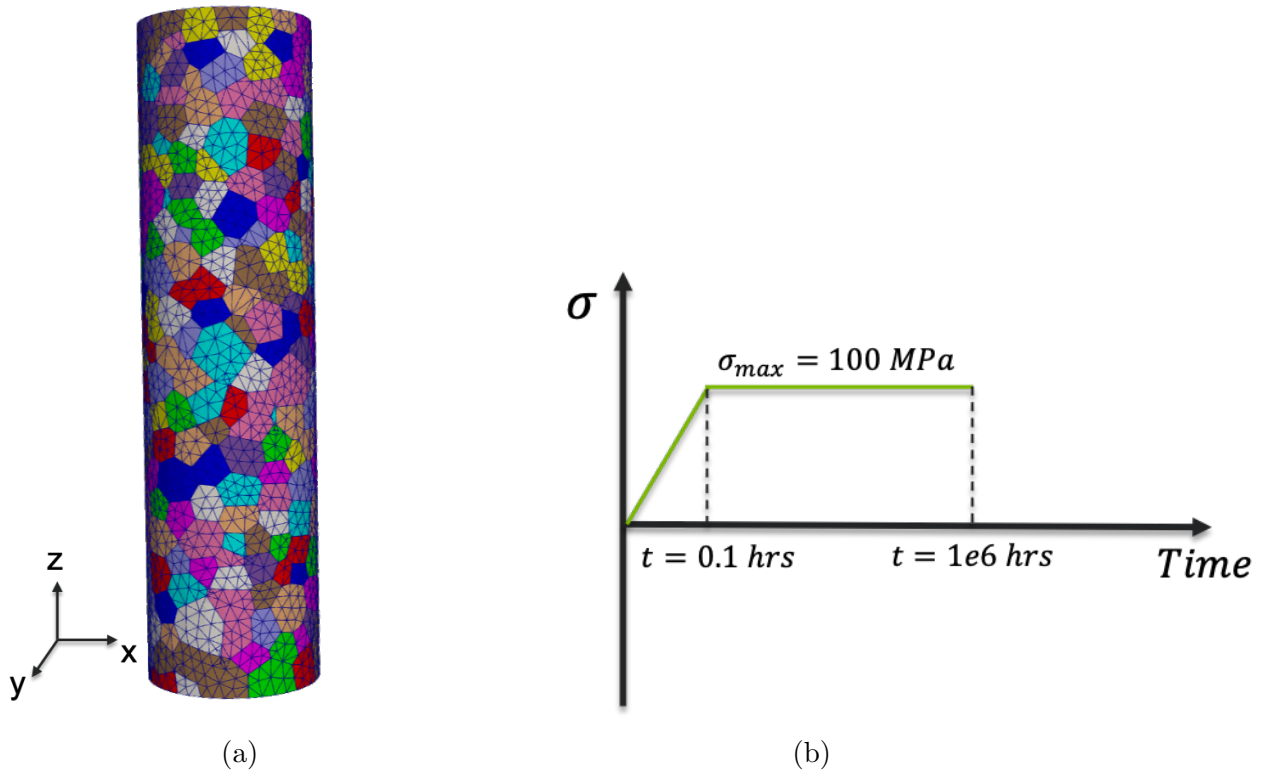
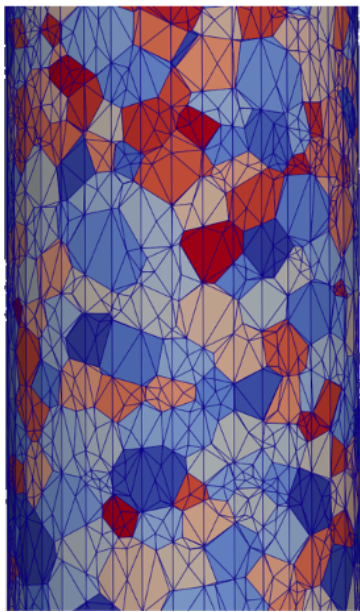
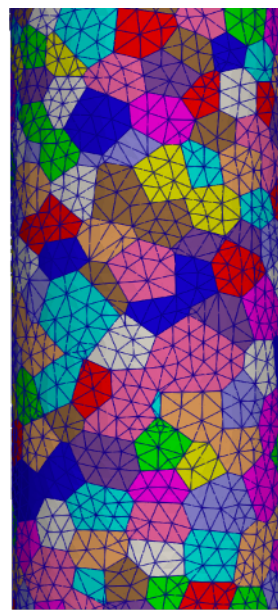


Figure 4.1: (a) A representative meshed microstructure used in the present work. This microstructure has 1000 grains and ~ 1.5 M elements. (b) Schematic of the creep loading conditions applied on the microstructure. Maximum stress of 100 MPa was used across all the simulations.



(a)



(b)

Figure 4.2: (a) Finite element mesh resulting from the old method using Gmsh and (b) mesh resulting from the updated method (using Coreform Trelis).

4.2 Results and Analysis

Many large scale microstructures were subjected to creep loading conditions to study the effect of free surfaces. Figure 4.3 shows the various geometries used in the study. All the microstructures were taken to tertiary creep, with the goal of identifying the effects of free surfaces, role of curvature in the microstructure geometry and also to establish the role the boundary conditions play in the strain partitioning. A summary of associated creep curves are shown in Figure 4.4. An important observation here is the early onset of tertiary creep stage in the cylinder geometry. It should be noted that “RVE” and “Cube” represent the cubic microstructures, with block periodic boundary conditions and free surfaces, respectively. The resulting strain distribution plot for the full cylinder microstructure is shown in Figure 4.5 (a). Also shown in Figure 4.5 (b) is the damage metric for the interfacial damage model (D) in the grain boundary cavitation model, which is defined as the ratio between the cavity half radius, a , and cavity half spacing, b :

$$D = \frac{a}{b} \quad (4.1)$$

Further, the material parameters used for the crystal plasticity are shown in Table 4.1. The parameters were described in more detail in an earlier report using this model [6].

Table 4.1: The material parameters used in the present simulations.

Symbol	Description	Value	Units
E	Young’s modulus	150,000	MPa
ν	Poisson’s ration	0.285	unitless
τ_{sat}	Isotropic saturation	12	MPa
τ_0	Isotropic initial value	40	MPa
$\dot{\gamma}_0$	Reference slip rate	9.55×10^{-8}	unitless
n	Flow exponent	12	unitless
β	Slope of the sigmoid ramp	2	unitless
c	Critical value of damage parameter	100	unitless

It is seen from the distribution of D in 4.5 (b) that there is no evidence of a clear crack path through the microstructure resulting from the creep simulation. Instead, the damage appears to be distributed across the microstructure somewhat homogeneously. This observation is consistent across different microstructures.

Furthermore, the average Poisson’s ratio is compared for the different simulated geometries in Figure 4.6. There is an observation of potentially two disparate mechanisms operating between the different types of microstructures. Specifically, the cylinder microstructures tend to have significantly larger values of average Poisson’s ratio in comparison with the cubic geometries. This observation is further investigated in the next section.

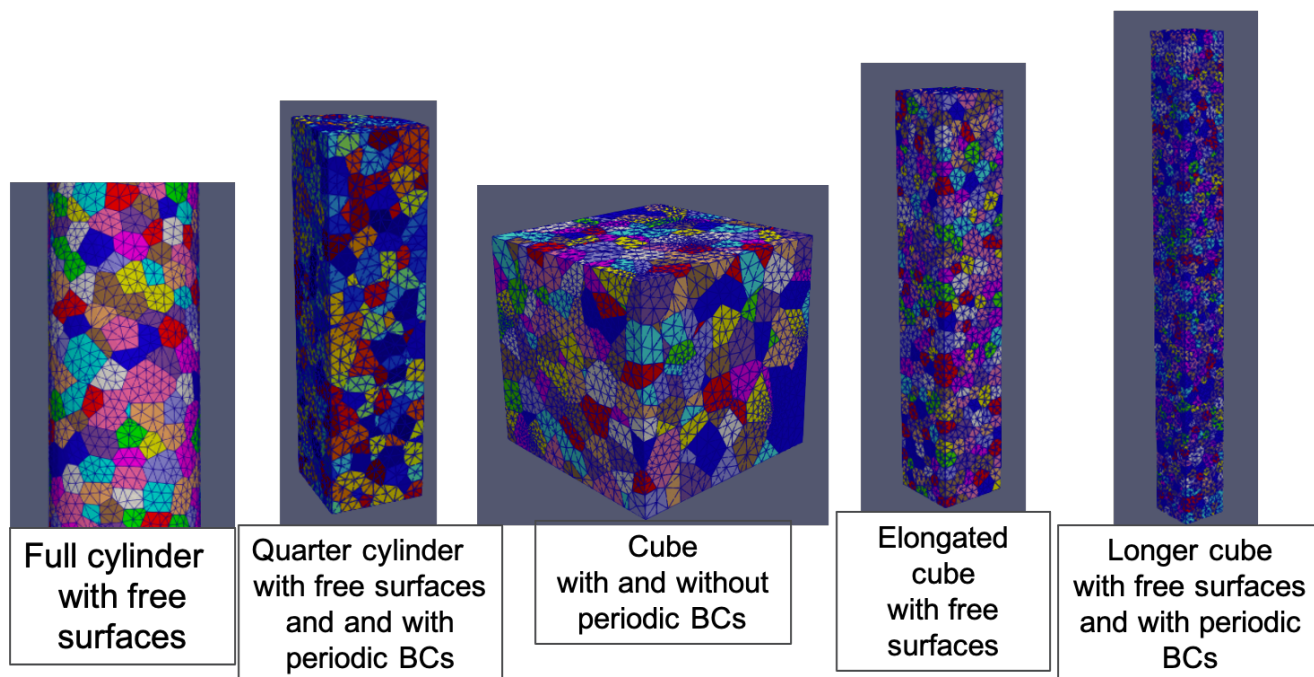


Figure 4.3: The geometries simulated in the present study: full cylinder, quarter cylinder, cube, and cube geometries with larger aspect ratios.

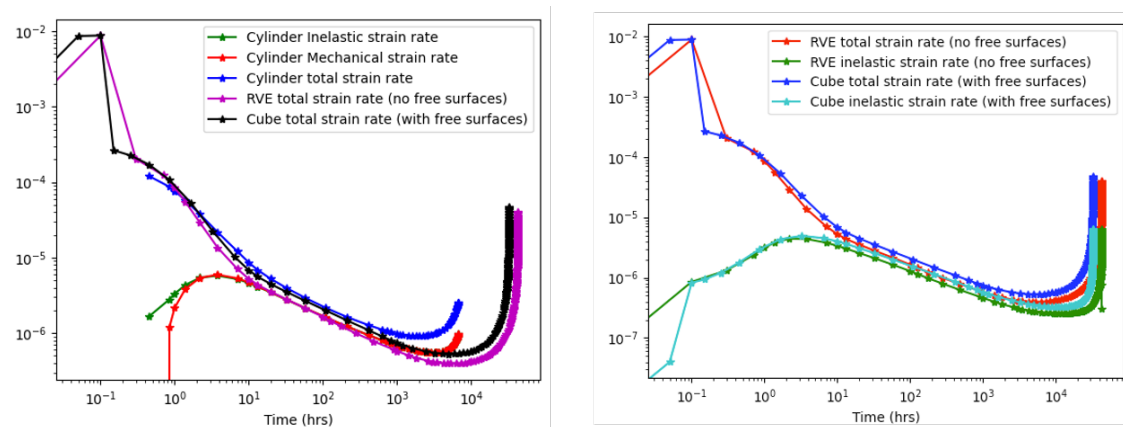


Figure 4.4: A compilation of the creep curves for the different geometries simulated are shown here. The cylinder microstructure reaches tertiary creep earlier than the cubic microstructure.

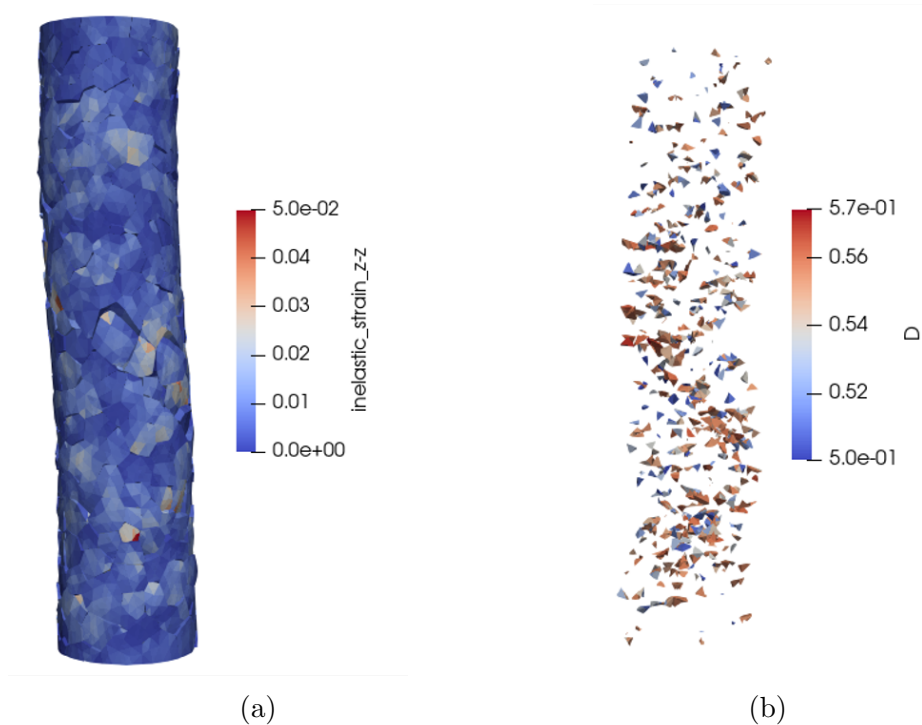


Figure 4.5: (a) The inelastic strain distribution at the end of the creep simulations in the full cylinder microstructure. (b) The corresponding distribution of the damage metric D in the cylinder microstructure.

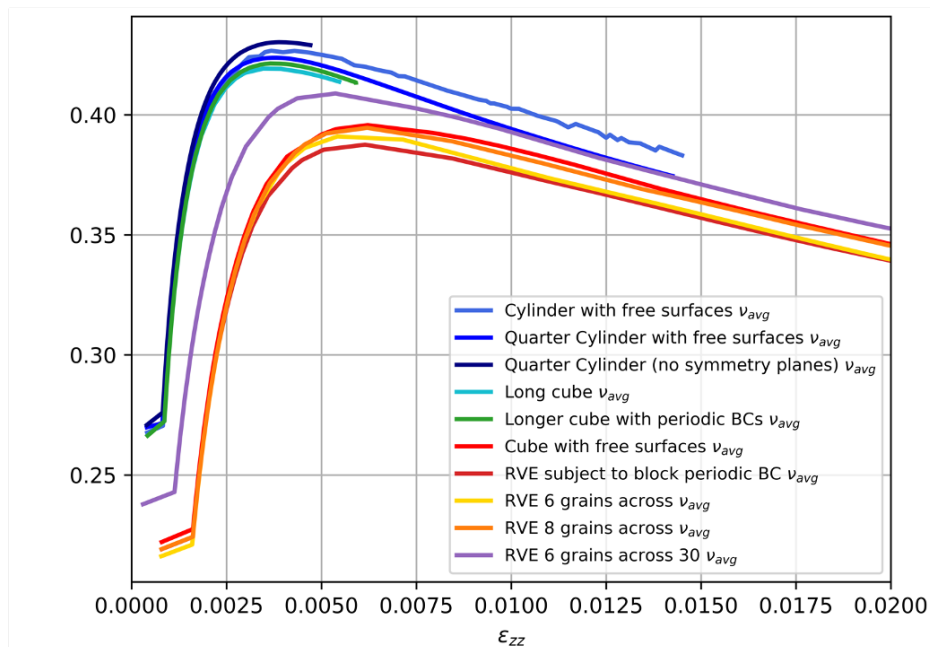


Figure 4.6: Comparison of average Poisson's ratio for the different simulated geometries.

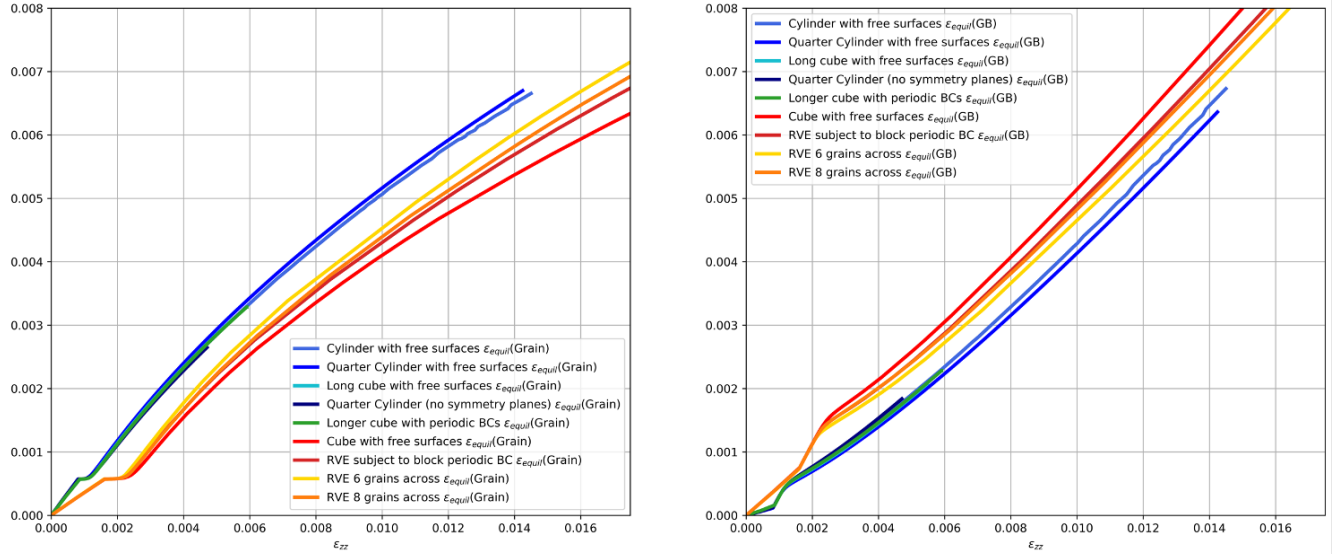


Figure 4.7: Comparison of the equivalent strains partitioned between the grain and grain boundaries (GB) for the different microstructures simulated.

Strain Partitioning

The strain partitioning between (a) the grain interiors and the grain boundaries and (b) grain boundary sliding and grain boundary cavitation/opening are compared in Figures 4.8, 4.9 and 4.10. It is again observed here that there might be two different active mechanisms between the geometries with curvature (full cylinder and quarter cylinder microstructures) and without curvature (cubic microstructures). Surprisingly, the boundary conditions do not appear to be a significant contributor to these differences in strain partitioning.

Summary

This section demonstrates the capabilities for full-field simulations of creep and creep-fatigue in MOOSE now available. This particular study focuses on the effect of the actual geometry of a test sample or component versus the more conventional periodic representative volume element cells often used in multiscale simulations of creep and fatigue processes. This study should be seen as an example of the new capabilities now in the process of being merged into the mainline MOOSE tensor mechanics module: the ability to run very large-scale simulations representing complex microstructural processes over realistic volumes of material. The new solid and interface mechanics kernels make simulations of this type possible.

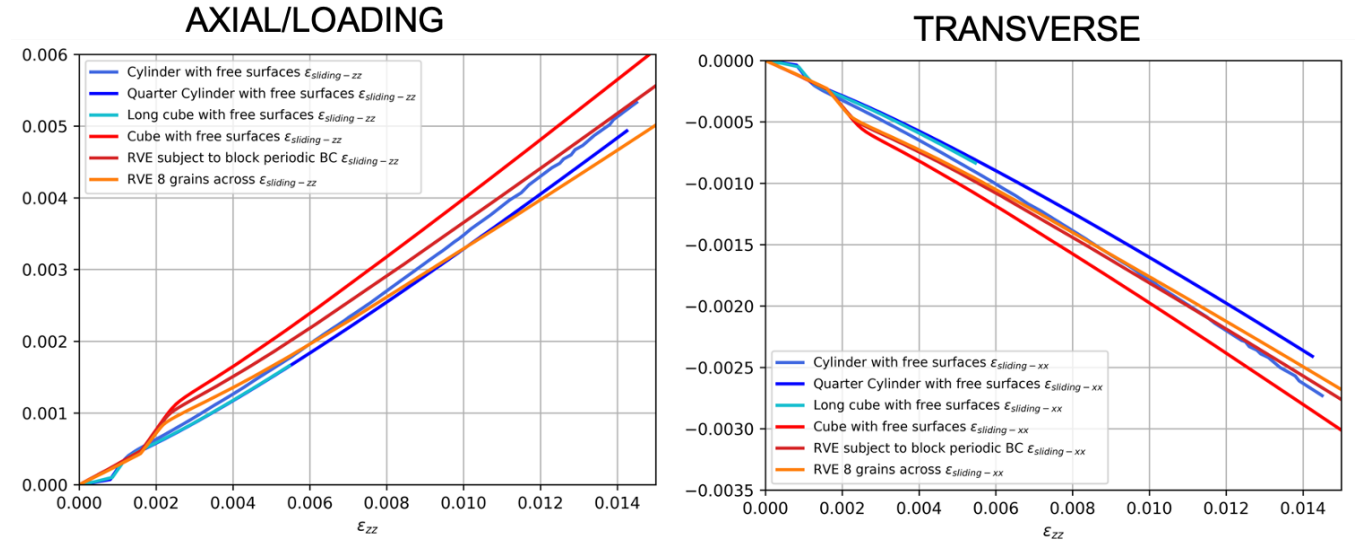


Figure 4.8: Comparison of the grain boundary sliding strains partitioned between the axial and transverse directions for the different microstructures simulated.

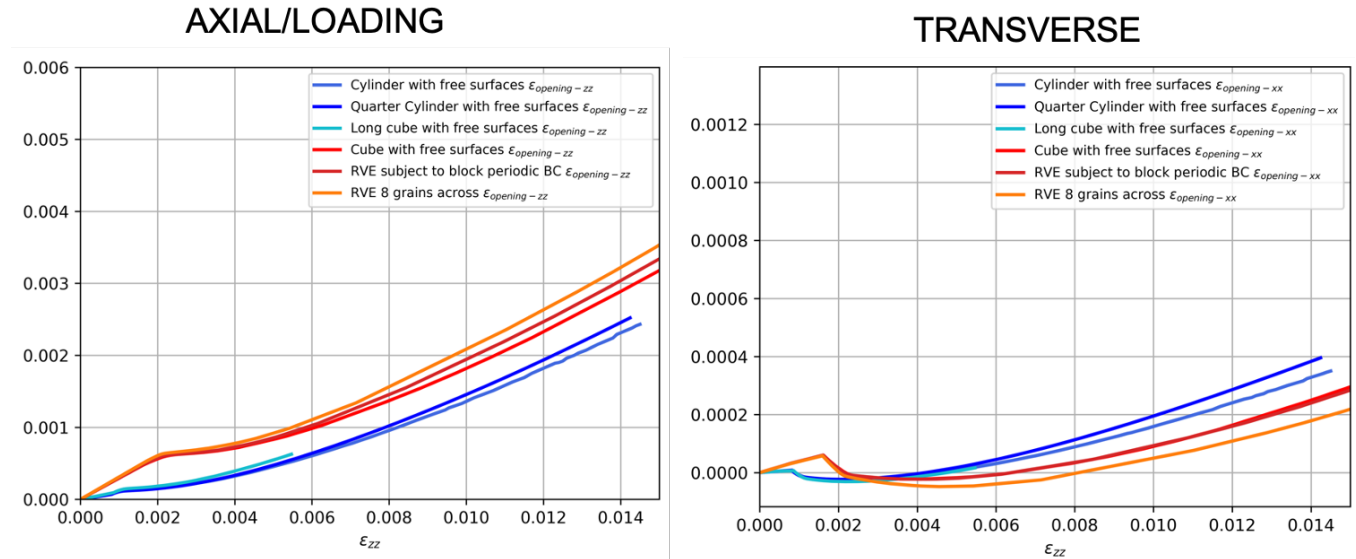


Figure 4.9: Comparison of the grain boundary opening strains partitioned between the axial and transverse directions for the different microstructures simulated.

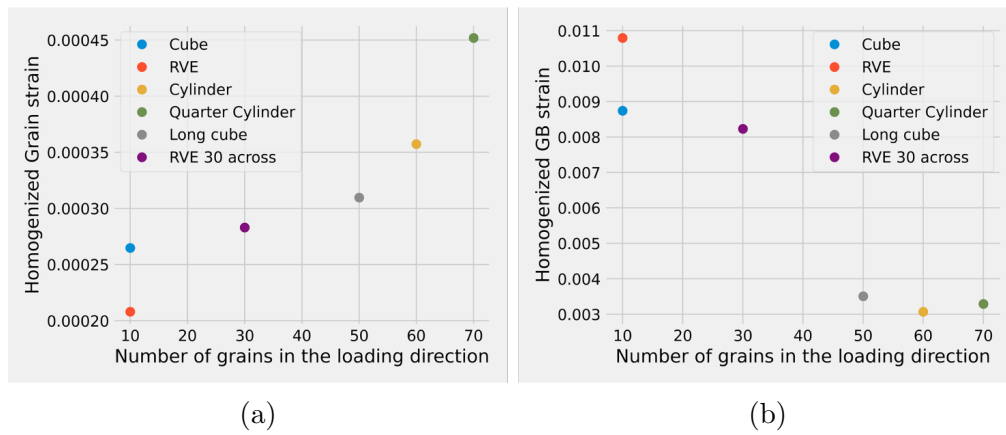


Figure 4.10: (a) Homogenized strains (homogenized over the entire domain of the microstructure) partitioned between the grain and grain boundaries (GB) for the different microstructures simulated.

5 Conclusions and Future Work

This report describes integrating new systems for solid and interface mechanics into MOOSE. The solid mechanics systems rewrites much of the basic implementation of the tensor mechanics module, providing a new kernel system with the goal of providing the exact Jacobian for stabilized, large-deformation solid mechanics simulations. This improvement in turn reduces the amount of time required to run large scale, large deformation simulations, for example crystal plasticity simulations of high temperature materials, by reducing the number of required nonlinear iterations required to find accurate, converged solutions.

The new interface-cohesive formulation provides the ability to insert interface material models along 2D interfaces in 3D models (or 1D interfaces in 2D models or 0D interfaces in 1D models). These types of models can be used to represent variety of material interface mechanics, including degradation and failure caused by creep cavitation on grain boundaries in crystal plasticity simulations.

Several components of these new frameworks were developed in previous MOOSE Apps. This report describes their integration to the main MOOSE framework, as well as several improvements beyond past work aimed at improvement the numerical performance of the systems and their ease of use:

- A new material system with interfaces for implementing a wide variety of solid mechanics constitutive models. This new system is compatible with the new solid mechanics kernels but also provide a means, via objective stress integration, to tie existing MOOSE material models into the new mechanics system.
- Intercompatible updated and total Lagrangian solid mechanics formulations.
- Stabilization for linear hexahedral elements for nearly-incompressible problems, including the exact linearization for large deformations.
- A new kinematic formulation for the interface model system that improves the performance of models for situations where interfaces undergo large rotations.
- A common interface material system with options for history independent and incremental, history-dependent models.

We have proposed both the solid mechanics and interface mechanics systems for incorporation into mainline MOOSE through two pull requests – #17475 for the solid mechanics system and #17157 for the interface-cohesive system. At the time of writing both pull requests are under review, but we hope to have both merged into mainline MOOSE before the publication of this report.

There is some work left to complete for the new solid mechanics system, specifically:

- The addition of axisymmetric and spherical version of the kernel, so that it can completely replace the existing tensor mechanics kernels.
- Integration of the current MOOSE material system with the new kernels, via the objective integration/small strain model
- Integration or reimplement of the generalized plane strain capability.

The last of these items can be easily accomplished using the homogenization strain system – it is in effect a specialized constraint of this type. None of these deficiencies in the new tensor mechanics system are critical – as demonstrated here, the current framework is sufficient to run the problems of interest.

Beyond this reintegration work, a few additional features would make running large scale CPFEM calculations easier or faster:

- Stabilization for linear tetrahedral elements. Most complex microstructures require tet meshes. Currently we use second order, quadratic elements to avoid locking. Stabilized linear tets would allow us to resolve geometric features at reduced computational cost, enabling larger-scale simulations.
- Improvements to the parallel scaling of the linear solver (HYPRE/BoomerAMG for large problems). Currently, the size of the CPFEM simulation is limited by the scaling of the linear solver. Long term, this is the critical issues to address in representing larger simulation volumes.

Acknowledgements

The research was sponsored by the U.S. Department of Energy, under Contract No. DEAC02-06CH11357 with Argonne National Laboratory, managed and operated by UChicago Argonne LLC. Programmatic direction was provided by the Nuclear Energy Advanced Modeling & Simulation Program of the Department of Nuclear Energy. The authors grateful acknowledge the support and direction of Ben Spencer, at Idaho National Laboratory, National Technical Director Chris Stanek, at Los Alamos National Laboratory, and Federal Manager Dave Henderson.

Bibliography

- [1] A. Rovinelli and M. C. Messner. Identify the influence of microstructure on mesoscale creep and fatigue damage. Technical Report ANL-20/49, Argonne National Laboratory, 2020. URL <https://www.osti.gov/biblio/1658575>.
- [2] Omar Nassif, Timothy J. Truster, Ran Ma, Kristine B. Cochran, David M. Parks, M. C. Messner, and T. L. Sham. Combined crystal plasticity and grain boundary modeling of creep in ferritic-martensitic steels: I. Theory and implementation. *Modelling and Simulation in Materials Science and Engineering*, 27(7), 2019. ISSN 1361651X. doi: 10.1088/1361-651X/ab359c.
- [3] M. C. Messner, Omar Nassif, Ran Ma, Timothy J. Truster, Kristine Cochran, David Parks, and T.-L. Sham. Combined crystal plasticity and grain boundary modeling of creep in ferritic-martensitic steels: II. The effect of stress and temperature on engineering and microstructural properties. *Modelling and Simulation in Materials Science and Engineering*, 27:075010, 2019.
- [4] MC Messner, A Rovinelli, DM Parks, and T-L Sham. Evaluation of statistical variation of microstructural properties and temperature effects on creep fracture of grade 91. Technical Report ANL-ART-143, Argonne National Lab.(ANL), Argonne, IL (United States), Lemont, IL, USA, 2018.
- [5] A. Rovinelli, M. C. Messner, Guosheng Ye, and T.-L. Sham. Initial study of notch sensitivity of grade 91 using mechanisms motivated crystal plasticity finite element method. Technical Report ANL-ART-171, Argonne National Laboratory, Lemont, IL, USA, 2020.
- [6] A. Rovinelli, M. C. Messner, and T.-L. Sham. Initial microstructural model for creep-fatigue damage in grade 91 steel. Technical Report ANL-ART-202, Argonne National Laboratory, Lemont, IL, USA, 2020.
- [7] Juan C Simo and Thomas JR Hughes. *Computational inelasticity*, volume 7. Springer Science & Business Media, 2006.
- [8] Zdeňek P Bažant and Jan Vorel. Energy-conservation error due to use of green–naghdi objective stress rate in commercial finite-element codes and its compensation. *Journal of Applied Mechanics*, 81(2), 2014.
- [9] MM Rashid. Incremental kinematics for finite element applications. *International Journal for Numerical Methods in Engineering*, 36(23):3937–3956, 1993.
- [10] T. J. R Hughes. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. New Jersey:Prentice-Hall, 1987.
- [11] EA de Souza Neto, D Perić, M Dutko, and DRJ Owen. Design of simple low order finite elements for large strain analysis of nearly incompressible solids. *International Journal of Solids and Structures*, 33(20-22):3277–3296, 1996.

- [12] Thomas JR Hughes. Generalization of selective integration procedures to anisotropic and nonlinear media. *International Journal for Numerical Methods in Engineering*, 15(9):1413–1418, 1980.
- [13] M Danielsson, DM Parks, and MC Boyce. Three-dimensional micromechanical modeling of voided polymeric materials. *Journal of the Mechanics and Physics of Solids*, 50(2): 351–379, 2002.
- [14] X-P Xu and A Needleman. Void nucleation by inclusion debonding in a crystal matrix. *Modelling and Simulation in Materials Science and engineering*, 1(2):111, 1993.
- [15] Mohsen Khajeh Salehani and Nilgoon Irani. A coupled mixed-mode cohesive zone model: An extension to three-dimensional contact problems. *arXiv preprint arXiv:1801.03430*, 2018.
- [16] Kyoungsoo Park and Glaucio H Paulino. Computational implementation of the ppr potential-based cohesive model in abaqus: Educational perspective. *Engineering fracture mechanics*, 93:239–262, 2012.
- [17] Yi Chao Chen and Lewis Wheeler. Derivatives of the stretch and rotation tensors. *Journal of Elasticity*, 32(3):175–182, 1993. ISSN 03743535. doi: 10.1007/BF00131659.
- [18] Argonne National Laboratory. NEML: the Nuclear Engineering material Model Library. <https://github.com/Argonne-National-Laboratory/neml>, 2019.
- [19] Cody J. Permann, Derek R. Gaston, David Andrš, Robert W. Carlsen, Fande Kong, Alexander D. Lindsay, Jason M. Miller, John W. Peterson, Andrew E. Slaughter, Roy H. Stogner, and Richard C. Martineau. MOOSE: Enabling massively parallel multiphysics simulation. *SoftwareX*, 11:100430, 2020. ISSN 2352-7110. doi: <https://doi.org/10.1016/j.softx.2020.100430>. URL <http://www.sciencedirect.com/science/article/pii/S2352711019302973>.
- [20] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.
- [21] Coreform. <http://www.coreform.com>.



Applied Materials Division

Argonne National Laboratory
9700 South Cass Avenue, Bldg. 212
Argonne, IL 60439

www.anl.gov



Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC